

Rochester Institute of Technology

## RIT Digital Institutional Repository

---

Theses

---

2010

### A Competitive gaussian model and linear regression technique for modeling MST neuron receptive field responses for primate perception of self-motion

Jeffrey Robble

Follow this and additional works at: <https://repository.rit.edu/theses>

---

#### Recommended Citation

Robble, Jeffrey, "A Competitive gaussian model and linear regression technique for modeling MST neuron receptive field responses for primate perception of self-motion" (2010). Thesis. Rochester Institute of Technology. Accessed from

This Thesis is brought to you for free and open access by the RIT Libraries. For more information, please contact [repository@rit.edu](mailto:repository@rit.edu).

**A Competitive Gaussian Model and Linear Regression  
Technique for Modeling MST Neuron Receptive Field  
Responses for Primate Perception of Self-Motion**

APPROVED BY

SUPERVISING COMMITTEE:

---

Dr. Roger Gaborski, Supervisor

---

Dr. Richard Zanibbi, Reader

---

Dr. Peter Anderson, Observer

**A Competitive Gaussian Model and Linear Regression  
Technique for Modeling MST Neuron Receptive Field  
Responses for Primate Perception of Self-Motion**

by

**Jeffrey Robble, B.S.**

**THESIS**

Presented to the Faculty of the  
Golisano College of Computer and Information Sciences  
Rochester Institute of Technology  
in Partial Fulfillment  
of the Requirements  
for the Degree of

**Master of Science**

**Rochester Institute of Technology**

June 2010

## Acknowledgments

I wish to thank Jennifer Ekblaw for her unconditional love, patience, and support throughout the last year, and both Dr. Charles Duffy and Dr. William Page for conducting visual experiments on Rhesus monkeys and collecting the neuronal firing rate response data which forms the foundation of this thesis research.



## **Abstract**

# **A Competitive Gaussian Model and Linear Regression Technique for Modeling MST Neuron Receptive Field Responses for Primate Perception of Self-Motion**

Jeffrey Robble, M.S.

Rochester Institute of Technology, 2010

Supervisor: Dr. Roger Gaborski

This thesis presents a linear regression technique applied to non-linear features extracted from a scaled Gaussian mixture model (SGMM) to describe the receptive field (RF) behavior of neurons in the Medial Superior Temporal region of the brain of Rhesus monkeys which viewed visual stimuli on a projector screen. The stimuli consist of simple and complex combinations of planar, circular, and radial motion and neuron responses were recorded in spikes/second. It is our goal to understand the neuronal responses of the primates to these stimuli and to create a model capable of predicting how the RFs in the primates' neurons respond to novel stimuli simulating the effects of self-motion. The SGMM is trained in stages using a competitive algorithm where a speciated genetic algorithm competes against a custom greedy algorithm.

The regions of each subject’s visual field which produce the most active neuron responses are identified as hotspot regions for that neuron. We strongly believe that the non-linear interaction between RFs can be explained by the relationship between hotspot and non-hotspot region behavior based on the direction of stimulus motion within each region. Our results show that the SGMM is capable of reasonably modeling and predicting firing rate response values for simple visual stimuli for a moderate number of data points, and that the linear model is capable of very accurately modeling and predicting firing rate response values for more complex stimuli for a small number of data points.

# Table of Contents

<b>Acknowledgments</b>	<b>iii</b>
<b>Abstract</b>	<b>iv</b>
<b>List of Tables</b>	<b>ix</b>
<b>List of Figures</b>	<b>xi</b>
<b>Chapter 1. Introduction</b>	<b>1</b>
<b>Chapter 2. Background</b>	<b>5</b>
2.1 Physiology of the Visual System . . . . .	5
2.2 Receptive Field Alignment Theory . . . . .	9
2.3 Gaussian Derivative Model . . . . .	10
<b>Chapter 3. Primate Experiments</b>	<b>13</b>
3.1 Experimental Procedure . . . . .	13
3.2 Visual Stimuli . . . . .	15
3.3 Regional Interaction Effect . . . . .	20
3.4 Opponent Organization . . . . .	23
3.5 Overlapping Gradients Hypothesis . . . . .	26
<b>Chapter 4. Previous Work</b>	<b>32</b>
4.1 Dual Gaussian Model . . . . .	32
4.2 Dual Gaussian Fitness Functions . . . . .	35
4.3 Dual Gaussian Genetic Algorithm . . . . .	36
4.4 Doubles Template Matching . . . . .	40

<b>Chapter 5. Methodology</b>	<b>43</b>
5.1 Scaled Gaussian Mixture Model Approach . . . . .	43
5.2 Multivariate Gaussian Distributions . . . . .	45
5.3 Expectation Maximization Consideration . . . . .	50
5.4 Probability Function and Directional Covariation Considerations . .	54
5.5 Training in Stages . . . . .	58
5.6 Training via Speciation . . . . .	60
5.7 Greedy Training Algorithm . . . . .	65
5.8 Competitive Training Algorithm . . . . .	69
5.9 Feature Space Considerations for Doubles and Flow Stimuli . . . . .	70
5.10 Linear Regression Technique . . . . .	71
5.11 Feature Selection . . . . .	74
5.12 Feature Sets . . . . .	79
5.13 System Design . . . . .	83
<b>Chapter 6. Results</b>	<b>87</b>
6.1 Probability Function and Directional Covariation Test Case . . . . .	87
6.2 Competitive Algorithm Applied to Singles Data . . . . .	93
6.3 Competitive Algorithm Applied to Doubles Data . . . . .	100
6.4 Linear Regression Technique Applied to Flow Data . . . . .	105
6.5 Flow Data Linear Model Feature Weight Analysis . . . . .	111
6.6 Linear Regression Technique Applied to Doubles Data . . . . .	121
<b>Chapter 7. Discussion</b>	<b>127</b>
7.1 Physiological Realism . . . . .	127
7.2 Dynamic Gain Theory . . . . .	130
<b>Chapter 8. Conclusion</b>	<b>134</b>
8.1 Summary . . . . .	134
8.2 Future Work . . . . .	137
<b>Appendices</b>	<b>140</b>

Appendix A.	Minimum, Maximum, and Base Firing Rates by Neuron	141
Appendix B.	Singles Stimuli Numbering Scheme	142
Appendix C.	Doubles Stimuli Numbering Scheme	143
Appendix D.	Additional Results for Competitive Algorithm Applied to Singles Data	146
Appendix E.	Additional Results for Competitive Algorithm Applied to Doubles Data	149
Appendix F.	Additional Results for Linear Regression Technique Applied to Flow Data	154
Appendix G.	Additional Flow Data Linear Model Feature Weight Analysis	157
Appendix H.	Feature Set Combinations Used for Flow Data Linear Model Feature Weight Analysis	161
Appendix I.	Additional Results for Linear Regression Technique Applied to Doubles Data	167
Appendix J.	Summary of Results by Neuron	170
Bibliography		172
Vita		176

## List of Tables

5.1	Hotspot segment features. . . . .	74
5.2	Non-hotspot segment features. . . . .	76
5.3	Type 1 feature sets. . . . .	80
5.4	Type 2 feature sets. . . . .	80
5.5	Type 3 feature sets. . . . .	80
6.1	Neuron 712R02 GA singles training results for PDF models with directional covariation. . . . .	90
6.2	Neuron 712R02 GA singles training results for PDF models with no directional covariation. . . . .	90
6.3	Neuron 712R02 GA singles training results for CDF models with directional covariation. . . . .	91
6.4	Neuron 712R02 GA singles training results for CDF models with no directional covariation. . . . .	91
6.5	CA singles prediction results. . . . .	95
6.6	Neuron 819R64 CA doubles subspace prediction results. . . . .	103
6.7	CA doubles prediction results. . . . .	103
6.8	LR flow prediction results using combination of all set features. . . .	106
6.9	LR flow prediction results using best segmental combination of set features. . . . .	106
6.10	LR flow prediction results using best exhaustive combination of set features. . . . .	107
6.11	Best combination of T3-FS4 feature weights for LR flow data prediction. . . . .	109
6.12	T3-FS4 feature indices. . . . .	115
6.13	LR doubles prediction results using combination of all set features. .	122
6.14	LR doubles prediction results using best segmental combination of set features. . . . .	122
6.15	LR doubles prediction results using best exhaustive combination of set features. . . . .	123

6.16	Best combination of T3-FS1 feature weights for LR doubles data prediction. . . . .	124
6.17	Best combination of T3-FS4 feature weights for LR doubles data prediction. . . . .	125
A.1	Minimum, maximum, and base firing rates by neuron. . . . .	141
B.1	Singles Oconfig. stimuli numbering scheme. . . . .	142
B.2	Singles Xconfig. stimuli numbering scheme. . . . .	142
C.1	Doubles Xconfig. stimuli numbering scheme part 1. . . . .	143
C.2	Doubles Xconfig. stimuli numbering scheme part 2. . . . .	144
C.3	Doubles Oconfig. stimuli numbering scheme. . . . .	145
E.1	Neuron 712R02 CA doubles subspace prediction results. . . . .	149
E.2	Neuron 819R09 CA doubles subspace prediction results. . . . .	150
E.3	Neuron 819R10 CA doubles subspace prediction results. . . . .	150
E.4	Neuron 819R32 CA doubles subspace prediction results. . . . .	151
G.1	T3-FS4 feature indices. . . . .	157
H.1	Neuron 712R02 best combinations of T3-FS4 features used for flow data linear model feature weight analysis. . . . .	162
H.2	Neuron 819R09 best combinations of T3-FS4 features used for flow data linear model feature weight analysis. . . . .	163
H.3	Neuron 819R10 best combinations of T3-FS4 features used for flow data linear model feature weight analysis. . . . .	164
H.4	Neuron 819R32 best combinations of T3-FS4 features used for flow data linear model feature weight analysis. . . . .	165
H.5	Neuron 819R64 best combinations of T3-FS4 features used for flow data linear model feature weight analysis. . . . .	166
J.1	Results summary. . . . .	170
J.2	Best combinations of T3-FS4 features used for doubles and flow data prediction. . . . .	171

# List of Figures

2.1	Optic tract through visual system. . . . .	6
2.2	Regions of the visual cortex. . . . .	7
2.3	Receptive field types. . . . .	9
2.4	Example of receptive field motion detection. . . . .	10
2.5	Gaussian derivative model motion detection example. . . . .	12
3.1	Rhesus monkeys and projector screen segments. . . . .	14
3.2	Examples of singles stimuli. . . . .	15
3.3	Screen segment configuration positions. . . . .	16
3.4	Examples of doubles stimuli. . . . .	18
3.5	Flow stimuli. . . . .	19
3.6	Example regional interaction effect for neuron 819R09. . . . .	21
3.7	Example regional interaction effect for neuron 712R02. . . . .	22
3.8	Example of opponent organization for neuron 819R64. . . . .	24
3.9	Example RF composed of two overlapping gradients. . . . .	28
3.10	Example RF responses. . . . .	30
4.1	Example dual Gaussian modeling an on-center off-surround RF. . . . .	33
4.2	GA crossover. . . . .	37
4.3	Example of doubles template construction. . . . .	42
5.1	Multivariate Gaussian space representation. . . . .	48
5.2	Multivariate Gaussian distribution in (x, y, dir.) space. . . . .	49
5.3	Overlapping multivariate Gaussian distributions. . . . .	49
5.4	Examples of Gaussians with and without directional covariation. . . . .	57
5.5	Staged training procedure. . . . .	59
5.6	Speciated training procedure. . . . .	63
5.7	Greedy training procedure. . . . .	68



5.8	Example of hotspot features. . . . .	77
5.9	Example of doubles pairing response features. . . . .	78
5.10	Possible LR system for doubles and flow prediction. . . . .	84
5.11	LR system for doubles prediction. . . . .	85
5.12	LR system for flow prediction. . . . .	85
5.13	CA system for hierarchical doubles prediction. . . . .	86
6.1	Example singles data GA test set. . . . .	88
6.2	Neuron 712R02 GA singles prediction MSE training performance. . .	92
6.3	Neuron 712R02 GA singles prediction MSE test performance. . . .	92
6.4	Neuron 819R32 CA singles prediction. . . . .	96
6.5	Neuron 819R32 CA singles training history. . . . .	97
6.6	Neuron 819R32 SGMM Gaussians in (x, y, dir) feature space. 3-D view.	98
6.7	Neuron 819R32 SGMM Gaussians in (x, y, dir) feature space. 2-D views.	99
6.8	Neuron 819R64 CA doubles prediction. . . . .	104
6.9	Neuron 712R02 LR flow prediction using best T3-FS4 combination. .	108
6.10	Neuron 712R02 LR flow prediction hotspot feature weights for best T3-FS4 combinations. . . . .	112
6.11	Neuron 712R02 LR flow prediction non-hotspot feature weights for best T3-FS4 combinations. . . . .	113
6.12	Neuron 712R02 LR flow prediction feature weight correlations for best T3-FS4 combinations. . . . .	116
6.13	Neuron 712R02 LR flow prediction feature weight correlation visual- ization for best T3-FS4 combinations. . . . .	117
6.14	Neuron 819R64 LR flow prediction feature weight correlations for best T3-FS4 combinations. . . . .	119
6.15	Neuron 819R64 LR flow prediction feature weight correlation visual- ization for best T3-FS4 combinations. . . . .	119
6.16	Neuron 712R02 LR doubles prediction using best T3-FS4 combination.	123
7.1	Physiological production of MST neuron response using dynamic gain factors. . . . .	133
D.1	Neuron 712R02 CA singles prediction. . . . .	146
D.2	Neuron 819R09 CA singles prediction. . . . .	147

D.3	Neuron 819R10 CA singles prediction. . . . .	147
D.4	Neuron 819R64 CA singles prediction. . . . .	148
E.1	Neuron 712R02 CA doubles prediction. . . . .	151
E.2	Neuron 819R09 CA doubles prediction. . . . .	152
E.3	Neuron 819R10 CA doubles prediction. . . . .	152
E.4	Neuron 819R32 CA doubles prediction. . . . .	153
F.1	Neuron 819R09 LR flow prediction using best T3-FS4 combination. .	154
F.2	Neuron 819R10 LR flow prediction using best T3-FS4 combination. .	155
F.3	Neuron 819R32 LR flow prediction using best T3-FS4 combination. .	155
F.4	Neuron 819R64 LR flow prediction using best T3-FS4 combination. .	156
G.1	Neuron 819R09 LR flow prediction feature weight correlations for best T3-FS4 combinations. . . . .	158
G.2	Neuron 819R09 LR flow prediction feature weight correlation visual- ization for best T3-FS4 combinations. . . . .	158
G.3	Neuron 819R10 LR flow prediction feature weight correlations for best T3-FS4 combinations. . . . .	159
G.4	Neuron 819R10 LR flow prediction feature weight correlation visual- ization for best T3-FS4 combinations. . . . .	159
G.5	Neuron 819R32 LR flow prediction feature weight correlations for best T3-FS4 combinations. . . . .	160
G.6	Neuron 819R32 LR flow prediction feature weight correlation visual- ization for best T3-FS4 combinations. . . . .	160
I.1	Neuron 819R09 LR doubles prediction using best T3-FS4 combination.	167
I.2	Neuron 819R10 LR doubles prediction using best T3-FS4 combination.	168
I.3	Neuron 819R32 LR doubles prediction using best T3-FS4 combination.	168
I.4	Neuron 819R64 LR doubles prediction using best T3-FS4 combination.	169

# Chapter 1

## Introduction

Understanding how the visual system of the mammalian brain functions on a physiological level can lead to innovative applications in the field of computer vision. For example, we may be able develop a biologically-inspired model that will allow a mobile robot to determine its self-motion (speed, orientation, and heading) by processing a live video stream. Many physics-based models employ kinematics, dynamics, positional and rotational geometry, and estimate velocities by tracking pixels frame-by-frame. Mammals do not perform such mathematical calculations to determine their self-motion, and therefore can be said to operate much more efficiently than robots employing such physics-based models. The neurological operations of the mammalian brain and underlying physiological behavior of the mammalian visual system are capable of determining self-motion very quickly with little effort and great accuracy. Therefore, a biologically-inspired model may be more desirable than a physics-based model. Based on our understanding of receptive field (RF) behavior of neurons located in the Medial Superior Temporal (MST) region of the mammalian brain, we can develop a computational model of RF activation suitable for machine learning, which may allow for the recognition, classification, and prediction of self-motion when deployed in intelligent agents [18].

Simulating the physiological processes of the mammalian visual system in software is a step towards creating man-made entities capable of realistic behavior analogous to that of human beings. A model that is capable of producing the neuronal firing rates of the MST region of the mammalian brain can potentially communicate with models simulating other regions of the brain responsible for higher-level cognition. It may even be possible to develop a cybernetic device employing such a model to interact with a real mammalian brain. A surgically implanted device may be able to aide or take the place of defective or degraded regions of the brain, such as those affected by Alzheimer’s disease. A common symptom of Alzheimer’s disease is sensory degradation, affecting the subject’s perception of spatial orientation and self-motion [15]. By developing a computational model of the MST region we can advance our understanding of the how the brain interprets visual sensory information in regards to self-motion. In turn this may lead to discoveries in the medical fields that can help diagnose, treat, and prevent this problem.

It is well understood that each RF in the MST region responds to visual stimulation moving along a specific direction in the 360° frontal view plane. A major point of interest is the theory that the MST region consists of multiple RFs aligned to respond to various orientations of motion and that these RFs interact in non-linear ways. For example, an excitatory response in one RF for rightward motion may elicit an inhibitory response in another RF for upward motion. Note that neuron responses are measured in terms of firing rates in spikes/second. It is unknown how many RFs compose any given neuron the

MST region, their relative sizes and positions, and how they interact with each other. Modeling the RFs within the MST region is non-trivial because there are many degrees of freedom, and each variability is highly dependent on the others.

The regions of the subject’s visual field which produce the most active neuron responses are identified as hotspot regions for that neuron. We strongly believe that the non-linear interaction between RFs can be explained by the relationship between hotspot and non-hotspot region behavior. Specifically, the direction of motion within the hotspot regions have a major impact on the overall firing rate produced by a neuron for any given stimulus. The overall firing rate is also strongly affected by the relationship between the direction of motion within hotspot regions compared to the direction of motion within non-hotspot regions. The relationship between the directions of motion within any pairing of non-hotspot regions is not as significant as the relationship between the directions of motion within a hotspot and non-hotspot region pairing. Additionally, we strongly believe that opposing directions of motion within the hotspot regions have inverse effects on the overall firing rate. For example, if rightward motion within the hotspot region produces an excitatory response then leftward motion within the hotspot region is likely to produce an inhibitory response; however, the response values are not simply equal and opposite, rather the magnitude of each response may vary drastically. This is known as opponent organization [20].

We take two different approaches to modeling RFs. The first approach

employs a competitive algorithm (CA) where a speciated genetic algorithm (GA) competes against a custom greedy algorithm (GRA) while training a scaled Gaussian mixture model (SGMM) composed of overlapping Gaussian distributions. The GA attempts to preserve innovative model solutions by dividing the different types of models into species and having models within the same species compete with each other for survival. The second approach employs a linear regression (LR) technique with non-linear features designed to extract inherent magnitudinal and directional information from the neuronal firing rate data. Various combinations of feature types and their positions within the RF feature space are considered.

This research in part extends that of Chen-Ping Yu, who applied a model composed of dual Gaussian distributions to model RFs in the MST region [17] [22]. Yu’s dual Gaussian model (DGM) restricts the position of each RF within the RF space to 1 of 9 segments; however, this constraint arises as a product of experimental procedure and is not supported by what we know about the physiology of the mammalian brain. The SGMM presented in this paper attempts to do away with this constraint and places no such restriction on the placement of RFs in the RF feature space. On the other hand, the LR technique operates on a discretized version of the RF feature space divided into the 9 conceptual segments in order to reveal non-linear interactions between regions in that space.

# Chapter 2

## Background

### 2.1 Physiology of the Visual System

The brain functions by sending electrical impulses between neurons at various firing rates. Activated neurons send electrical signals to other neurons, forming a complex communication network. When a visual stimulus is perceived by the eye, photoreceptors in the back of the retina are excited and fire electrical signals down the optic nerve to the Lateral Geniculate Nucleus (LGN), which acts as a hub and directs these signals to other regions of the brain, including the primary visual cortex [10]. The optic tract is identified in the cross-section of the human brain illustrated in Figure 2.1. The signals used to determine the motion of objects in the subject's environment and the subject's self-motion travel along the Dorsal stream and pass through the V1, V2, medial temporal (MT), and then the Medial Superior Temporal (MST) region. The V1 and V2 regions process color information, some directional information, and perform basic edge detection. The MT region recognizes where objects are located in the environment. After passing through these regions, the signals arrive at the MST region at the end of the Dorsal stream (MSTd), which processes directional and self-motion information [10]. These regions are illustrated in Figure 2.2.

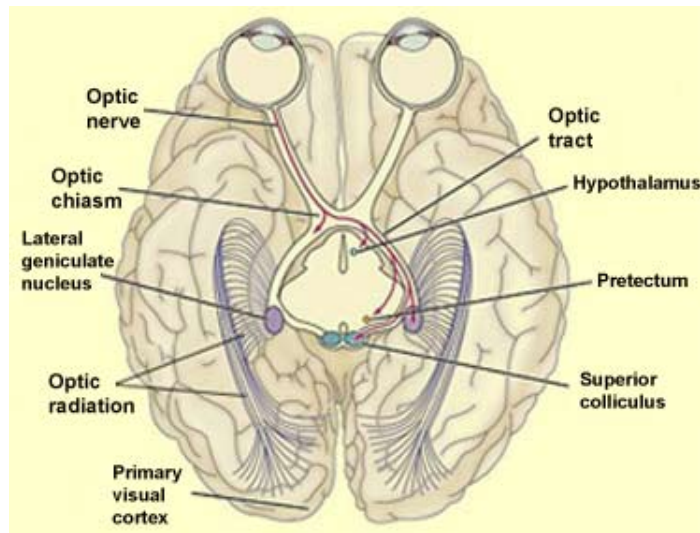
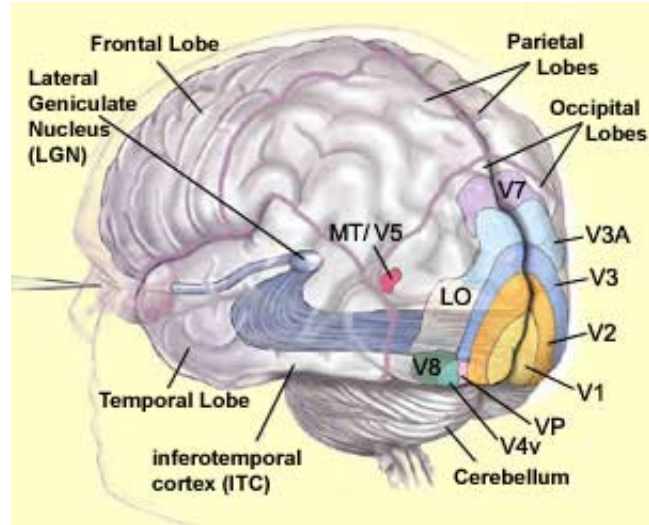


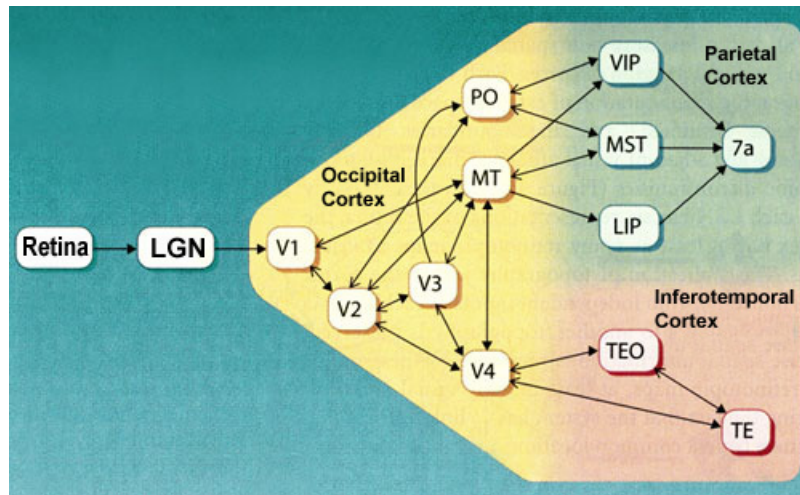
Figure 2.1: Optic tract along which electrical impulses travel from the retina to the LGN along the optic nerve and are then routed to the visual cortex. Image taken from <http://thebrain.mcgill.ca>.

The structure and operation of the non-human primate brain is similar to that of the human brain. Both have defined V1, V2, MT, and MST regions (among others) that perform similar behaviors and are believed to serve the same purpose. The vast majority of visual experiments performed on non-human primate subjects to date employ electrodes, or an array of electrodes, to measure the firing rates of specific neurons in the visual cortex and surrounding areas of the primate's brain. In Section 3.1 we discuss the electrode experiments performed by Dr. Duffy and Dr. Page, wherein the data which forms the foundation for this thesis research was recorded. On the other hand, the vast majority of visual experiments performed on human subjects employ functional magnetic resonance imaging (fMRI) to measure blood oxygenation





(a) Relative locations of V1, V2, and MT regions.



(b) Interconnections between regions.

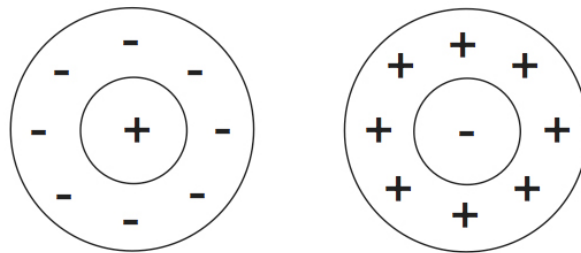
Figure 2.2: Regions of the visual cortex responsible for processing self-motion. Images taken from <http://thebrain.mcgill.ca>.

level dependent (BOLD) signals in the relevant regions of the human brain [2]. Measuring neuron firing rates using electrodes requires removing part of the subject's skull cap while performing fMRI measurements is a non-invasive procedure. For this reason human subjects are rarely subjected to electrode experiments.

The neuronal firing rate data measured using electrode techniques provide a very low-level view of the operation of the subject's brain. Such experiments measure neuron behavior in isolation of other neurons, or use an electrode array to measure the behavior of a patch of neurons. In comparison, the BOLD data measured using fMRI techniques provide a high-level view of the operation of the subject's brain. Such experiments measure changes in local blood flow, which indicate regions of activity within the brain [13]. The exact relationship between neuron activity and changes in BOLD signals is not known. Increases in BOLD signals do not always indicate excitatory neurons, rather, increases in BOLD signals may indicate neurons which produce firing rate responses that are significantly different from the passive state of the subject [2]. Such neurons may be excitatory or inhibitory. In order to make a more direct comparison between the behavior of the non-human primate brain compared to the human brain, and to identify regions of the brain in both groups which serve the same function, further fMRI experimentation must be performed on non-human primates. Recent advances in fMRI technology advocate parallel studies on human and non-human primate subjects using fMRI techniques [13] [9].

## 2.2 Receptive Field Alignment Theory

Ganglion cells are a cluster of bundled soma and dendrite structures capable of receiving electrical stimulation and integrating synaptic inputs. RFs are composed of ganglion cells located in the retina which respond to light to alter the firing rates of neurons in the visual system. The size of a RF is typically measured in degrees of the visual field. Large RFs respond to larger regions of the visual field when stimuli are presented within those regions. RFs can be categorized into two primary types: on-center off-surround and off-center on-surround [21]. When a visual stimulus overlaps the center of the an on-center off-surround RF an excitatory response occurs and the RF fires more than usual. When a visual stimulus overlaps the periphery of an on-center off-surround RF an inhibitory response occurs and the RF fires less than usual. Figure 2.3(a) shows an on-center off-surround RF. Off-center on-surround RFs work in the opposite way; they are inhibited when a stimulus overlaps the center of the RF and excited when a stimulus overlaps the periphery of the RF. Figure 2.3(b) shows an off-center on-surround RF.



(a) On-center off-surround. (b) Off-center on-surround.

Figure 2.3: Receptive field types. See description in text.

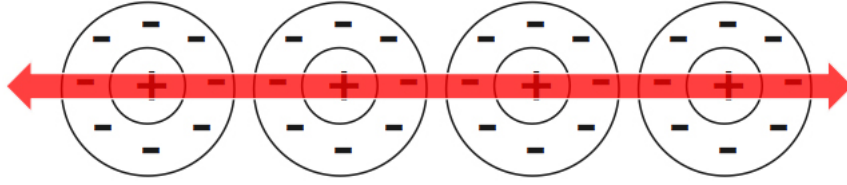


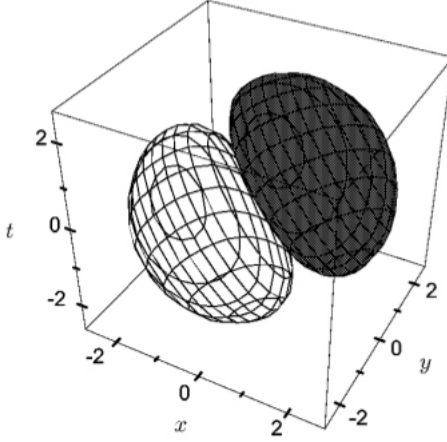
Figure 2.4: Example demonstrating how multiple horizontally aligned on-center off-surround RFs can be used to detect horizontal motion across the stimulus region denoted by the red arrow. This stimulus causes a large excitatory response because it overlaps the center of each RF. If the stimulus was shifted slightly upwards or downwards it would cause a less excitatory response.

These types of RFs can be aligned in different ways to determine the direction of stimulus motion and in turn the relative direction of the subject's self-motion. For example, if multiple on-center off-surround RFs are aligned horizontally they can be used to detect horizontal motion. This is shown in Figure 2.4.

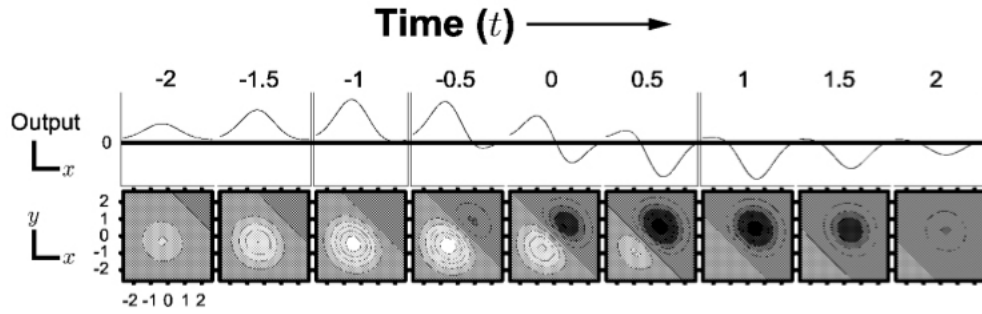
## 2.3 Gaussian Derivative Model

One of the most successful applications of the RF alignment theory is the Gaussian derivative model (GDM) developed by Young, et al., to detect motion in video [4]. Young employs Gaussian distributions to simulate the behavior of RFs. The first step of the process involves selecting a derivative form of a 3-D Gaussian in (x, y, time) space to use. The form of the Gaussian selected is based on the type of motion being detected and other properties of the video of interest. The next step is to sequentially convolve each time

frame of the GDM with each frame from the video of interest. Taking a slice of the model at a specific point along the time axis results in a 2-D plane in the  $(x, y)$  space that is used as a convolutional kernel for that time period. As each kernel is convolved with the respective frame of video, any object or edge moving in a similar direction as the GDM's orientation during that time period will respond with higher intensity output, therefore detecting motion in the video [4] [17]. An example is shown in Figure 2.5.



(a) Gaussian model field used for edge detection.



(b) RF profile as an edge moves from the upper right to lower left over time ( $-2t$  to  $2t$ ).

Figure 2.5: Gaussian derivative model motion detection example. The Gaussian model field shown in (a) was applied to a stimulus consisting of diagonal edge movement. The leftmost lobe is positive and the rightmost shaded lobe is negative. The RF profile plots in (b) show the  $(x, y)$  convolutional kernel at each time frame overlapping with the edge movement. Each kernel is simply a horizontal slice of the model shown in (a) along the corresponding time ( $t$ ) axis. The response output increased as the edge approached the threshold between the positive and negative lobes and decreased as the threshold was passed. Images taken from [4].

## Chapter 3

### Primate Experiments

#### 3.1 Experimental Procedure

At the University of Rochester Dr. Charles Duffy and Dr. William Page trained Rhesus Macaques (i.e. Rhesus monkeys), such as those shown in Figure 3.1(a), to sit in a chair and stare at a red dot in the center of a black rear-projection screen covering the central visual field ( $90^\circ \times 90^\circ$ ), as illustrated in Figure 3.1(b). The screen was conceptually divided into 3 rows and 3 columns, forming 9 segments for the purpose of showing stimuli. The segments were conceptual in the sense that no indication of the division was visible on-screen and the primate may have not been explicitly aware of the existence of the 9 segments. After the subject was anesthetized, trephine holes centered above the MSTd region of the subject's brain were cut through the skull cap. Bilateral recording cylinders (i.e. electrodes) were then centered above the holes to record neuronal firing rates. The behavior of each neuron was measured in isolation. An electrode array was not used. Scleral search coils [3] were implanted around the limbus of both eyes and a magnetic resonance technique was used to verify that the subject was staring at the fixation point [11] [12].

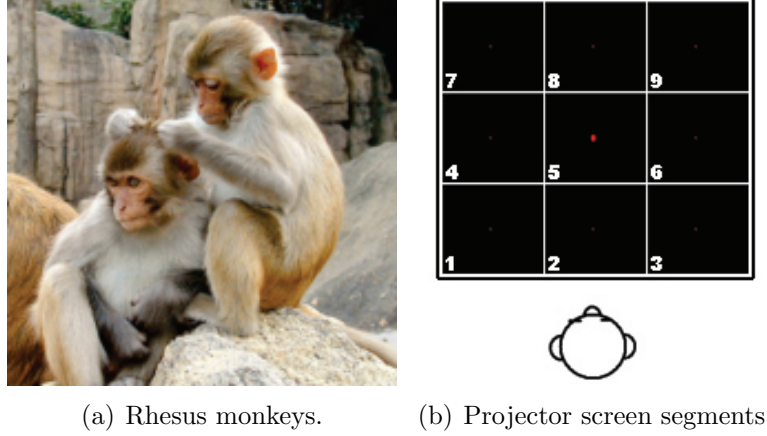


Figure 3.1: (a) Rhesus monkeys. Image taken from <http://www.cosmosmagazine.com>. (b) Diagram of a monkey staring at a fixation point on in the center of a projector screen divided into 9 conceptual segments. Each segment is numbered for reference. Numbers did not appear on the screen. Image taken from [17] and modified.

After preparing the subject, various visual stimuli were displayed on screen and neuronal firing rates were recorded for multiple neurons per stimulus. Additionally, base firing rate values were recorded when no stimulus was displayed on the screen. Firing rates were recorded for each stimulus over 52 neurons divided between 2 primate subjects. The recorded neurons for the first primate do not correspond to the recorded neurons for the second primate. It was difficult to determine the exact position of the neurons in each of the primates' MST regions during experimentation. To account for this the modeling techniques presented in this thesis consider each neuron independently of the others. Additionally, in this thesis research a subset of 5 neurons was selected for modeling purposes. The neurons selected exhibited non-trivial activity levels and what appeared to be non-random behavior that could be modeled reasonably well. Minimum, maximum, and base firing rate values per neuron are listed in Appendix A.



### 3.2 Visual Stimuli

Each stimulus consisted of 500 white dots of equal size, density, and speed, divided into one or more groups, each moving in a separate screen segment. Once any dot reached the terminating edge of a segment it would reappear at the originating edge in order to maintain uniform coverage within the segment. Three types of stimuli were displayed: singles, doubles, and optic flow. Singles stimuli consisted of a single group of dots moving in one of the 4 primary directions in one of 9 screen segments, resulting in 36 combinations. Examples of singles stimuli are shown in Figure 3.2.

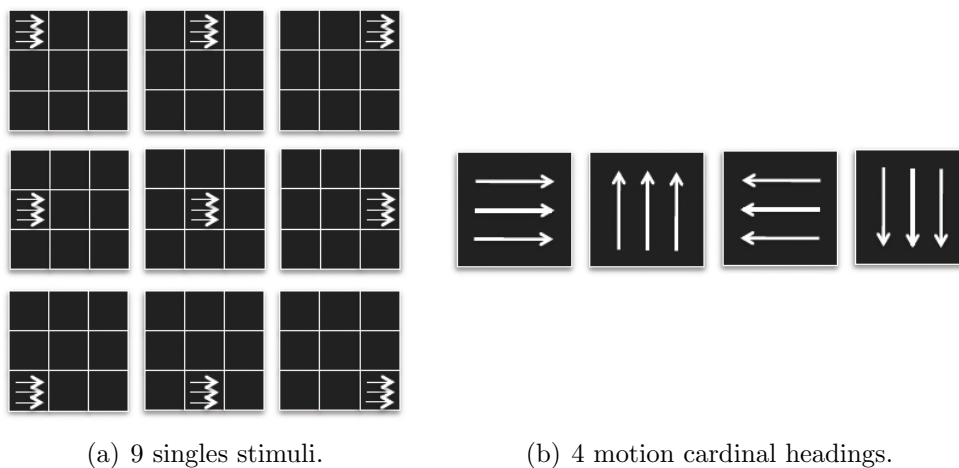


Figure 3.2: Examples of singles stimuli. Moving white dots appear in 1 of the 9 screen segments for each stimulus. (a) 9 stimuli for motion heading at  $0^\circ$ . (b) 4 cardinal motion headings for singles and doubles stimuli. Images based on those in [17] and [23].

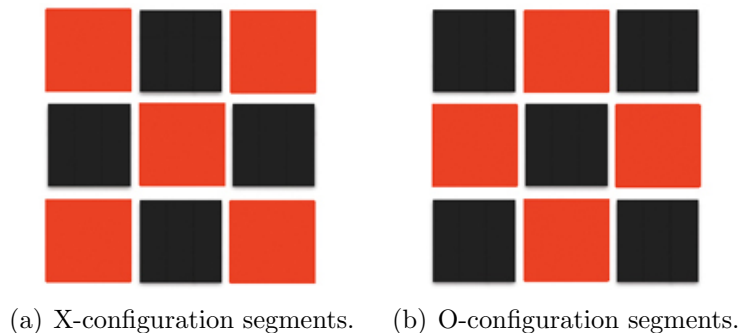


Figure 3.3: Screen segment configuration positions. Stimuli are shown in the red segments for each configuration.

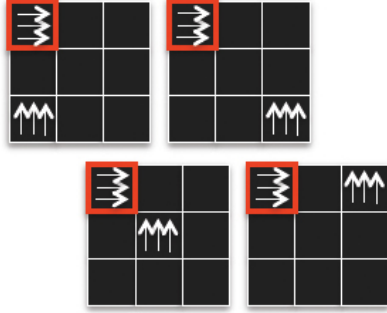
The order in which singles and doubles stimuli were presented alternated between screen segments in an X-configuration and O-configuration to prevent too much concentrated stimulation in one region of the RF space, which could have a negative effect on the experiment results. The X-configuration consists of the 4 segments in the corners of the screen and the one segment in the center of the screen, as shown in Figure 3.3(a). The O-configuration consists of the remaining 4 screen segments, as shown in Figure 3.3(b). The X-configuration screen segment for which the primate produced the most active response among all of the singles stimuli for any of the possible 4 motion directions for a given neuron was identified as the X-configuration “hotspot” segment for that neuron. The most active response is that which is the most different from the base firing rate value. Similarly, a separate O-configuration hotspot segment was determined for each neuron.

Doubles stimuli consisted of 2 groups of moving dots. One group was displayed in one of the hotspot segments and the other group was displayed in one of the non-hotspot segments for the given configuration. For some stimuli the direction of motion in the hotspot segment was the same as that for the non-hotspot segment. For other stimuli the direction of motion was different. Examples of doubles stimuli are shown in Figure 3.4. For both configurations all stimulus combinations of hotspot direction, non-hotspot direction, and non-hotspot segments were tested for the 4 cardinal directions, totaling 112 doubles stimuli for each neuron:

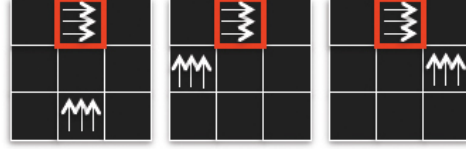
$$\begin{aligned}
& (4 \text{ hotspot motion headings}) * (4 \text{ non-hotspot motion headings}) * \\
& \quad (4 \text{ open non-hotspot X-configuration segments}) = 64 \text{ stimuli} \\
& (4 \text{ hotspot motion headings}) * (4 \text{ non-hotspot motion headings}) * \\
& \quad (3 \text{ open non-hotspot O-configuration segments}) = 48 \text{ stimuli} \\
& (64 \text{ X-configuration doubles stimuli}) + (48 \text{ O-configuration doubles stimuli}) \\
& \quad = 112 \text{ doubles stimuli}
\end{aligned}$$

The last type of stimuli consisted of moving white dots in 8 or 9 of the screen segments to simulate the effect of self-motion. Of the 16 optic flow stimuli, 8 of them correspond to planar motion, 2 correspond to circular motion, and 6 correspond to radial (inward/outward) motion. The discretized versions of the flow stimuli are shown in Figure 3.5.

The actual flow stimuli shown to the primate consisted of continuous fluid motion over the entire screen area. The direction of moving dots within any one of the 9 segments was not necessarily uniform. For example, the



(a) X-configuration segments.



(b) O-configuration segments.

Figure 3.4: Examples of doubles stimuli. Moving white dots appear in 2 of the configuration screen segments for each stimulus, one in the hotspot segment (red outline), and one in a non-hotspot segment. This example shows hotspot motion heading at  $0^\circ$  and non-hotspot motion heading at  $90^\circ$ . (a) X-configuration doubles stimuli where the upper-left segment is the hotspot. (b) O-configuration doubles stimuli where the upper-center segment is the hotspot. Images based on those in [23].

direction of motion within a corner segment for circular patterns differed up to  $90^\circ$  in order to form an arc of moving white dots. In order to simplify the data and model construction, the average direction within each of the segments is selected to represent motion within the segment as a whole. Thus, a diagonal direction is used to represent the motion in the corner segments of circular patterns. We refer to doubles and flow stimuli as complex stimuli (as compared to the simpler singles stimuli).

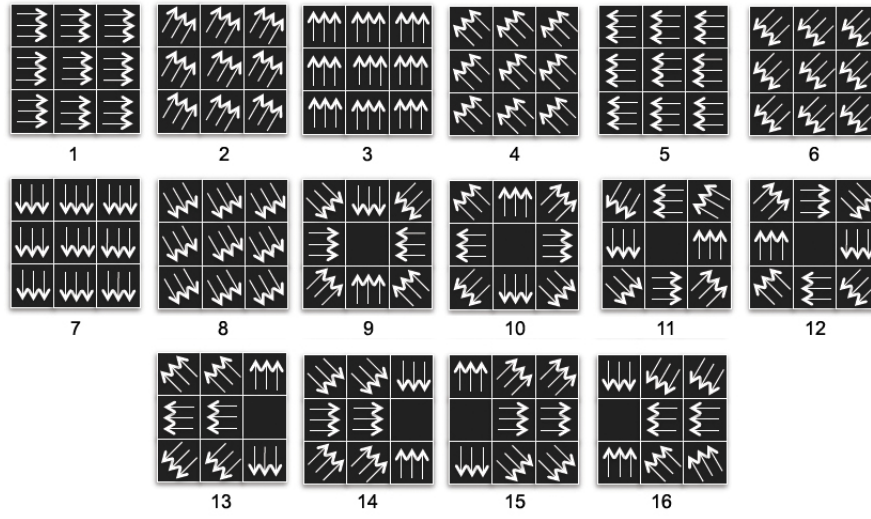


Figure 3.5: Flow stimuli. Stimuli 1 through 8 are planar, stimuli 11 and 12 are circular, and the rest are radial. Stimulus 10 simulates what would be perceived if the subject was to walk into the screen. Similarly, stimulus 9 simulates what would be perceived if the subject was to walk backwards while looking at the screen. Images based on those in [17] and [23].

### 3.3 Regional Interaction Effect

By analyzing the recorded doubles stimuli data, we know that when multiple groups of dots are displayed on screen the response of one region of the RF space can affect the response of another region by changing the response magnitude, inverting the response, or both. In order to better reflect what we know about the physiology of the primate visual system, it would be most desirable if our model takes into account this interaction effect of regional responses. Figure 3.6 shows a simple example of the interaction effect.

A doubles stimulus consists of a group of dots in a hotspot screen segment and a non-hotspot screen segment. Note that 2 hotspots are determined for each neuron, one for the segments in the X-configuration and one for the segments in the O-configuration. Duffy and Page identify a hotspot as the most responsive configuration segment for any one of the singles stimuli for a given neuron. If we inspect the recorded singles responses for a given direction of motion, such as those shown in Figure 3.6(a), and compare those responses to the responses of similar doubles stimuli, such as those shown in Figure 3.6(b), we may notice that the firing rates for corresponding singles and doubles stimuli change in a significant way. The example in Figure 3.6 shows that the addition of the hotspot stimulus to the singles stimulus increases the firing rate response for each of the stimuli. Another example in Figure 3.7 (for a different neuron and configuration) shows that the addition of the hotspot stimulus to the singles stimulus increases the firing rate response for some of the stimuli and decreases it for others. In both examples the effect that the

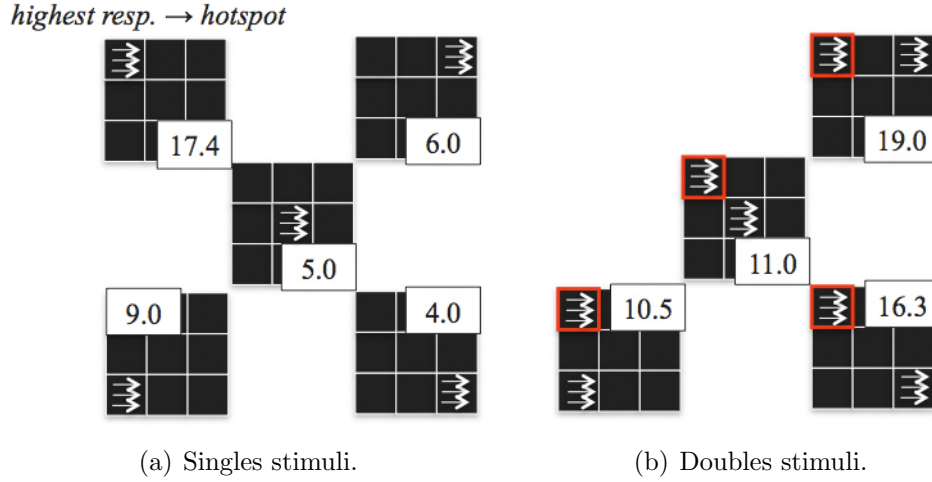


Figure 3.6: Example of regional interaction effect for neuron 819R09. (a) X-configuration singles stimuli for motion heading at  $0^\circ$ . Numbers indicate neuron firing rates in spikes/second. The neuron produces the highest firing rate when motion is shown in the upper left segment of the screen, so that segment becomes the hotspot. (b) X-configuration doubles stimuli for hotspot motion heading at  $0^\circ$  (outlined in red) and non-hotspot motion heading at  $0^\circ$ . Notice how all firing rates increase due to the addition of hotspot motion. Images inspired by those in [23].

addition of the hotspot segment to the singles stimuli has on the firing rate response magnitude is not clear. Also, the relationship between the direction of motion within the hotspot segment and direction of motion within the non-hotspot segment, and its effect on the firing rate response magnitude, is not clear. The effects appear to be non-linear.

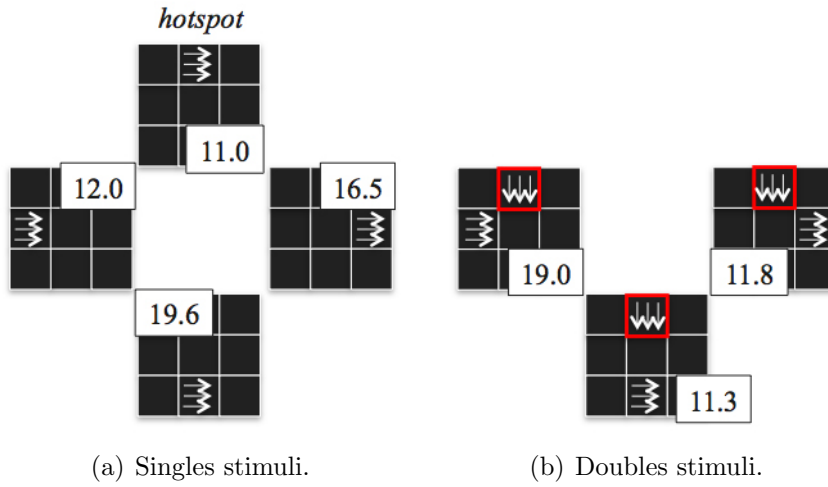


Figure 3.7: Example of regional interaction effect for neuron 712R02. (a) O-configuration singles stimuli for motion heading at  $0^\circ$ . The uppermost segment is selected as the hotspot because that is where the neuron produces the highest firing rate for singles stimuli (specifically for motion heading at  $90^\circ$ , which is not shown in this figure). (b) O-configuration doubles stimuli for hotspot motion heading at  $270^\circ$  and non-hotspot motion heading at  $0^\circ$ . Notice how some firing rates increase due to the addition of hotspot motion and others decrease. Images inspired by those in [23].



### 3.4 Opponent Organization

Opponent organization is a theory presented by Steinmetz, et al., that partially explains the regional interaction effect [20]. The theory is based on the assumption that for a given neuron and configuration the hotspot region has a direct impact on the firing rates generated for motion shown in combinations of hotspot and non-hotspot regions (such as double and flow stimuli). Specifically, one can make a general prediction about the firing rate that will be produced by the neuron based on the motion heading within the hotspot. Figure 3.8 shows a clear example of a neuron which exhibits opponent organization behavior.

Consider the top left plot shown in Figure 3.8. This plot is a visualization of 4 separate doubles stimuli. One of the stimuli consists of the center hotspot with a motion heading of  $0^\circ$  and the top left non-hotspot with a motion heading of  $0^\circ$ . This stimulus produced a firing rate of 14.5 spikes/second. Similarly, another stimulus consists of the center hotspot and bottom left non-hotspot, each with a motion heading of  $0^\circ$ . This stimulus produced a firing rate of approximately 14.3 spikes/second. The other two stimuli consist of the center hotspot and either the top right non-hotspot or lower right non-hotspot, respectively, which each produced a firing rate of 13.5 spikes/second.

Visualizing the data in this way allows us to make some observations by comparing stimuli consisting of the same hotspot and non-hotspot segments, the same non-hotspot motion headings, but opposing hotspot motion headings. Consider the top left plot again. Specifically, consider the stimulus

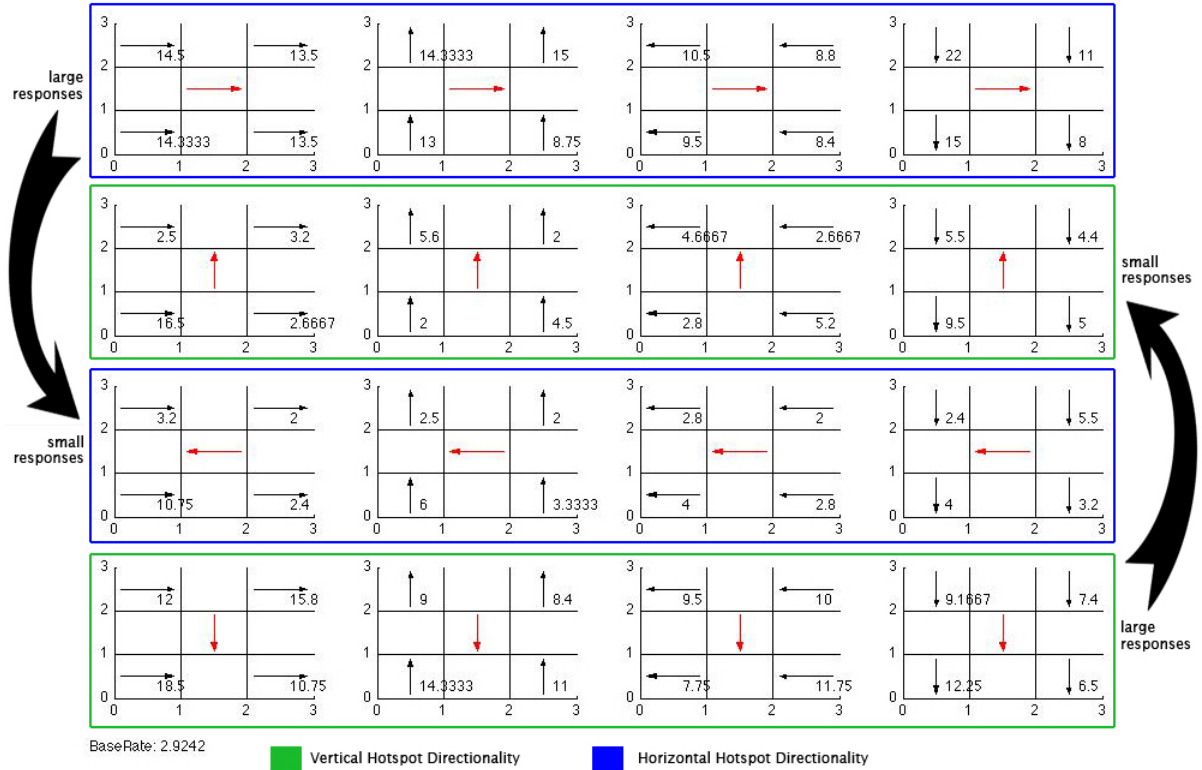


Figure 3.8: Example visualization of opponent organization for neuron 819R64. All 64 X-configuration doubles stimuli firing rates are shown. Each of the 16 plots shows the firing rate for 4 doubles stimuli. Each doubles stimulus consists of the red hotspot motion heading for the center hotspot segment and one of the other 4 non-hotspot motion headings shown in each plot. All non-hotspot motion headings are the same down each column. All hotspot motion headings are the same across each row. When comparing doubles stimuli consisting of the same hotspot and non-hotspot segments, all of the firing rates for stimuli with a hotspot motion heading of  $0^\circ$  are larger than those with a hotspot motion heading of  $180^\circ$ . The same is true for stimuli with a hotspot motion heading of  $270^\circ$  when compared to those with a hotspot motion heading of  $90^\circ$ .

consisting of the hotspot and top left non-hotspot segment within that plot, which produced a firing rate of 14.5 spikes/second. Now consider the plot that is two plots below that one. Specifically, consider the stimulus consisting of the hotspot and top left non-hotspot segment within that plot, which produced a firing rate of 3.2 spikes/second. The only difference between these two stimuli is the hotspot motion heading, which is  $0^\circ$  in the first stimulus and  $180^\circ$  in the second stimulus. The non-hotspot motion heading is  $0^\circ$  in both stimuli. The stimuli consisting of the hotspot and the bottom left non-hotspot segment in these two plots can be compared in a similar way, the first stimulus producing a firing rate of approximately 14.3 spikes/second and the second producing a firing rate of 10.75 spikes/second.

Essentially we are comparing doubles stimuli where the hotspot motion heading is  $180^\circ$  different. In other words, the hotspot motion heading between the stimuli is opposite. We observe that the firing rates for all of the comparable stimuli with hotspot motion headings of  $0^\circ$  are larger than those with a hotspot motion heading of  $180^\circ$ , and the firing rates for all of the comparable stimuli with hotspot motion headings of  $270^\circ$  are larger than those with a hotspot motion heading of  $90^\circ$  for this neuron. Other neurons may produce larger firing rates for hotspot motion headings of  $180^\circ$  versus  $0^\circ$  and  $90^\circ$  versus  $270^\circ$ . Additionally, some neurons do not exhibit this type of behavior at all.

In this example it is clear that the neuron has a directional preference for rightward and downward hotspot motion. For this reason we say that the neuron is directionality selective. If we were presented with a doubles stimulus

consisting of hotspot motion heading at  $0^\circ$  and non-hotspot motion heading at  $45^\circ$ , we could safely predict that it would produce a firing rate larger than that for a doubles stimulus consisting of hotspot motion heading at  $180^\circ$  and the same non-hotspot motion heading. Our ability to make this prediction plays a large role in the features we select for the LR technique discussed in Section 5.10.

### 3.5 Overlapping Gradients Hypothesis

Duffy and Wurtz present the overlapping gradients hypothesis to explain how overlapping “planar mechanisms might selectively produce responses to circular and radial stimuli” [12]. Specifically, Duffy and Wurtz state the following four inferences about the structure and behavior of RFs of MSTd neurons based on their experimentation:

1. RFs are made up of gradients such that planar motion over the center of the RF produces stronger responses than planar motion over the periphery of the RF. Concentric contours forming around the center of the gradient represent regions of equal response magnitude.
2. Each RF contains at least one excitatory gradient and one inhibitory gradient.
3. Each gradient can induce different response magnitudes.
4. Each gradient has its own directional preference.

Duffy and Wurtz state that each RF is composed of multiple gradients and it is the placement and directional preferences of the gradients which allow the subject to determine self-motion and the motion of objects in his environment. On the other hand, we have discussed a theory which explains how the alignment of multiple on-center off-surround and off-center on-surround RFs can be used to determine motion headings in Section 2.2. Conceptually, the overlapping gradients hypothesis and RF alignment theory are similar in that both state that multiple elements are responsible for determining motion headings. It is not clear whether or not those elements are RFs in and of themselves or those elements are gradients that compose a single RF.

According to Duffy and Wurtz, planar motion over a gradient region can either produce an inhibitory or excitatory response, which is in accordance to what is discussed in Section 2.2 and shown in Figure 2.4. Upward planar motion within the top, bottom, left, or right region of the visual field box shown in Figure 3.9 will cause a small inhibitory response because such regions overlap with the inhibitory gradient, which responds to upward motion, as indicated by the short dotted arrow pointing upwards. Since the gradient is circular, stimuli which share that same direction of motion and are positioned equal distances from the center of the gradient share the same response. These stimuli are illustrated in the fourth row down in Figure 3.10. Unlike the inhibitory gradient, which is responsive to motion in all four cardinal directions, the excitatory gradient is responsive only to rightward motion (as indicated by the long solid rightward arrow) and ignores all other directions of motion.

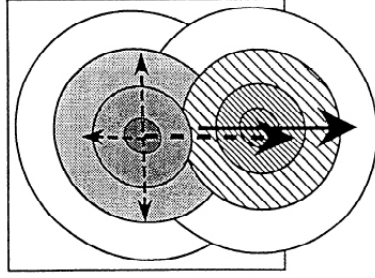


Figure 3.9: Example RF composed of two overlapping gradients. The box represents the region of the subject’s visual field in which stimuli are perceived. The left gradient is inhibitory and the right gradient is excitatory. Concentric circles represent regions of equal response magnitude. The dotted arrows represent the inhibitory gradient’s motion preferences for the four cardinal directions, especially for rightward motion. The longer the arrow, the stronger the preference. The long solid arrow represents the excitatory gradient’s large preference for rightward motion only. Image taken from [12].

Upwards planar motion within the center of the inhibitory gradient will cause a slightly larger inhibitory response than motion overlapping with the edge of the gradient. Downward planar motion within the center of the inhibitory gradient will cause the same response (note that the downward dotted arrow is the same length as the upward dotted arrow). Leftward planar motion within the center of the inhibitory gradient will cause the smallest inhibitory response (note the shorter length of the leftward dotted arrow) and rightward planar motion within the center of the inhibitory gradient will cause the largest inhibitory response (note the longer length of the rightward dotted arrow). Rightward planar motion within the right side of the box will overlap with the excitatory gradient and will cause a large excitatory response. Note that the excitatory gradient in Figure 3.9 extends outside of the boundary of the visual field box. This is acceptable because the size of the subject’s RF

may be larger than the subject's visual field, although the subject does not perceive or respond to stimuli outside of the visual field.

Note that the length of the solid rightward arrow is the same as the dotted rightward arrow in Figure 3.9. This indicates that the inhibitory gradient's preference for rightward motion is equal to the excitatory gradient's preference for rightward motion. Rightward planar motion between the centers of the two gradients will have an inhibitory response contribution and an excitatory response contribution. Summing the value of these contributions forms a single response with an approximate value of zero because of the equal and opposite response contributions of the two gradients. Figure 3.10 shows various stimuli and their relative responsive values for the RF shown in Figure 3.9. Complex motion within the entire RF can be determined by summing the individual response contributions produced by each of the gradients for the direction of motion overlapping those gradient regions. Discretizing a flow pattern into 9 segments, each corresponding to a region of the RF, makes this calculation much easier. In the next chapter we will discuss how each gradient can be represented by a Gaussian distribution.

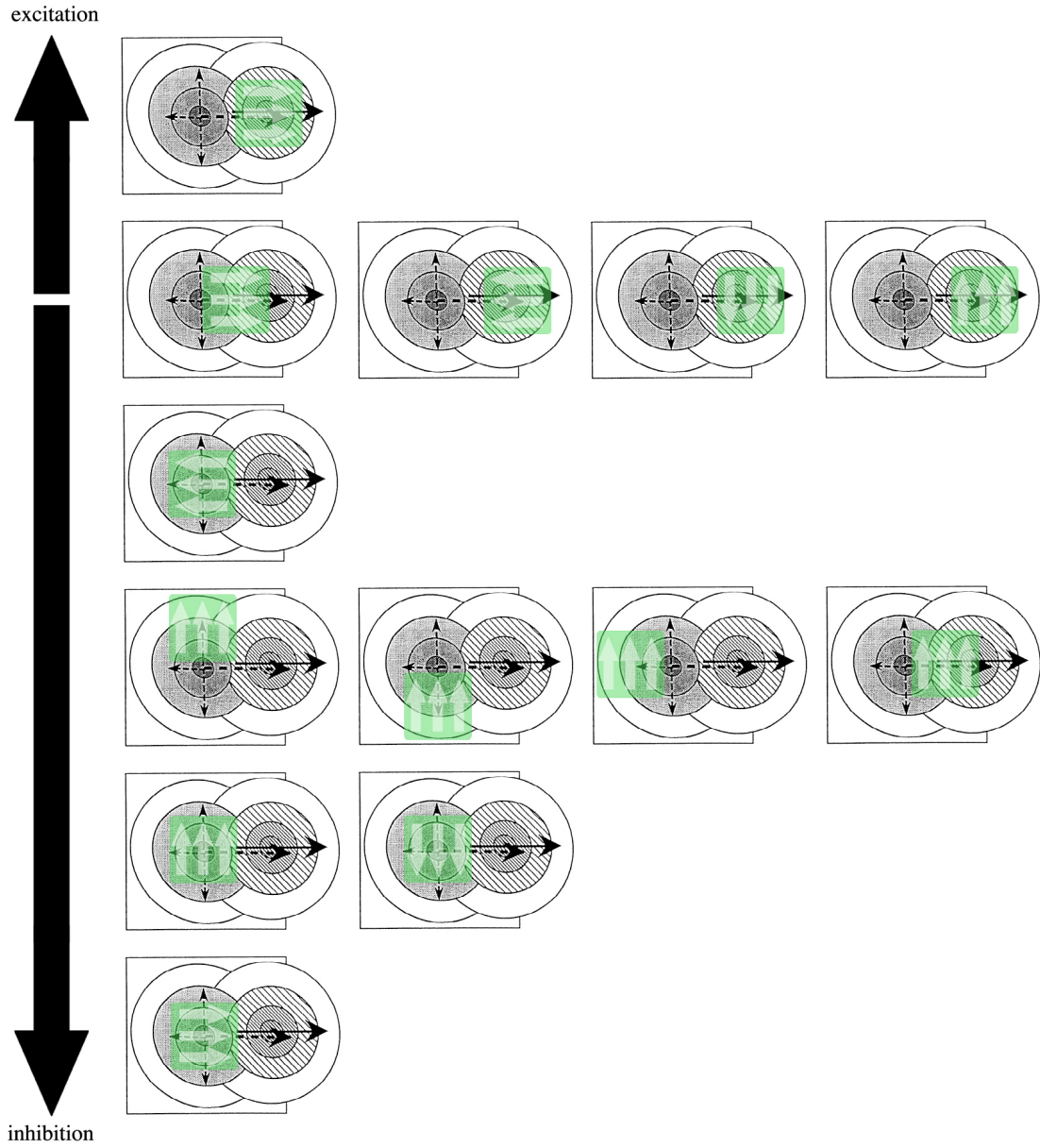


Figure 3.10: Example RF responses. Each visual field contains a different stimuli. The green overlay shows the stimuli direction of motion and region of overlap with the gradients in the RF. Stimuli in the same row share the same approximate level of response magnitude and polarity (excitatory or inhibitory). See descriptions of stimuli in text. Topmost and bottommost stimuli produce responses with similar magnitude but opposite polarity. Images taken from [12] and modified.



Row 1 (top): The excitatory gradient is very responsive to rightward motion.

Row 2: (leftmost) Since the excitatory and inhibitory gradients have an equal preference for rightward motion, a rightward stimulus that overlaps each gradient area by the same amount will result in a combined response of zero. (others) Non-rightward motion over the excitatory gradient will result in a response of zero because the excitatory gradient is only responsive to rightward motion.

Row 3: The inhibitory gradient is semi-responsive to leftward motion.

Row 4: (all) The inhibitory gradient is responsive to upward motion. The area of the gradient in which the stimulus overlaps is not important; however, the amount of gradient area overlapped is important, which is the same for all 4 examples shown in this row.

Row 5: (all) The inhibitory gradient is responsive to upward and downward motion. More of each stimulus overlaps with the gradient than the stimuli shown in row 4, therefore producing greater inhibitory responses.

Row 6 (bottom): The inhibitory gradient is very responsive to rightward motion.

# Chapter 4

## Previous Work

### 4.1 Dual Gaussian Model

Using the data collected by Duffy and Page, Yu developed a dual Gaussian model (DGM) to describe the neuronal behavior of the primates [17]. A Gaussian distribution can calculate the firing rate response of one of the 9 screen segments as a function of the direction of stimulus motion. Underlying Yu's work is the assumption that each of the 9 screen segments can be appropriately represented using an alignment of on-center off-surround or off-center on-surround RFs. The RF alignment theory states that the combination of these RFs is most responsive to one direction of motion and that the response can be excitatory or inhibitory. Yu models this by centering a Gaussian distribution around the most responsive direction, which is accomplished by setting the Gaussian mean ( $\mu$ ) parameter. Then Yu selects a polarity factor ( $P$ ) that will be multiplied by the Gaussian probability by to account for excitatory ( $P = 1$ ) and inhibitory ( $P = -1$ ) responses and a multiplicative constant ( $C$ ) to account for variations in neuronal response magnitude. The last parameter is the standard distribution ( $\sigma$ ) of the Gaussian, which accounts for how selective the RF is to variations in motion direction. The standard form of a normalized Gaussian distribution is shown in Equation 4.1.

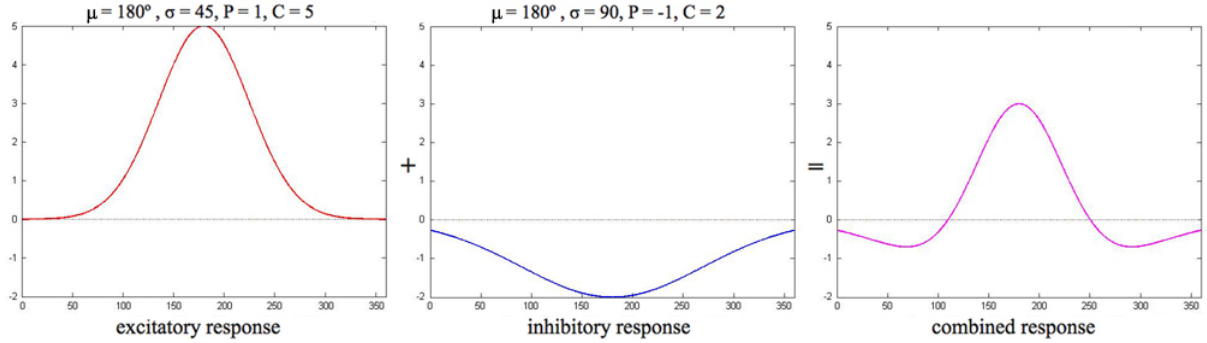


Figure 4.1: Example dual Gaussian modeling an on-center off-surround RF. Each plot shows the firing rate response value versus the direction of motion within the Gaussian's segment. The leftmost plot shows the excitatory Gaussian response output, the center plot shows the inhibitory Gaussian response output, and the rightmost plot shows the dual Gaussian response output, which is simply the summation of the excitatory Gaussian and inhibitory Gaussian responses.

$$G_{\mu,\sigma}(x) = e^{\frac{(x-\mu)^2}{2\sigma^2}} \quad (4.1)$$

To model the behavior of on-center off-surround RFs, Yu uses the combined effect of summing the output of two Gaussians. The on-center aspect requires an excitatory Gaussian with a small standard deviation and the off-surround aspect requires an inhibitory Gaussian with a larger standard deviation. By centering both Gaussians around the same mean Yu forms a function that responds with positive values for direction of motion close to the mean and negative values for directions of motion further away from the mean. An example is shown in Figure 4.1. A similar approach can be used to model the behavior of off-center on-surround RFs. This method of representing RFs using overlapping Gaussians is the driving principle behind the DGM. In general,

one Gaussian is used to model excitation responses and one Gaussian is used to model inhibition responses within the same segment. The response output by the dual Gaussians for a given screen segment,  $i$ , is shown in Equation 4.2. The variable subscripts refer to one of the two Gaussians within that segment.

$$response_{segment_i}(x) = C_{i_1} * G_{\mu_{i_1}, \sigma_{i_1}}(x) * P_{i_1} + C_{i_2} * G_{\mu_{i_2}, \sigma_{i_2}}(x) * P_{i_2} \quad (4.2)$$

In order to predict the response of a given stimulus, the dual Gaussian response for each of the 9 screen segments is calculated with respect to the direction of motion within each respective segment and then the segmental response contributions are summed. If there is no motion within a segment then that segment is inactive and contributes a response of 0 to the sum. The response calculated for a given stimulus,  $j$ , is shown in Equation 4.3. Note that Gaussians model the difference between the neuron firing rate for a given stimulus and the neuron base firing rate. Therefore, Equation 4.3 would return the base firing rate if no stimulus were shown on screen, as expected.

$$response_{model_j} = baserate + \sum_{i=1}^9 response_{segment_i} \quad (4.3)$$

Yu developed a genetic algorithm (GA) to train a DGM to fit the singles response data for a given neuron. Specifically, Yu's GA determines appropriate values for each of the 4 Gaussian parameters ( $\mu, \sigma, C, P$ ) for each of the Gaussian distributions in the system. Since there is one DGM for each of the 9 screen segments, there are 18 Gaussian distributions. Therefore,  $4 * 18 = 72$  parameter values per neuron must be determined. Yu chose a GA approach due to the complex interdependencies between these parameters.

## 4.2 Dual Gaussian Fitness Functions

Before we discuss the operation of Yu's GA, let us first discuss how Yu chose to calculate and compare the fitness between DGMs, where each DGM is a member of the GA population. The fitness of each member is determined using Equation 4.4 and Equation 4.5. Index  $j$  refers to one of the 36 singles stimuli.

$$Fitness_{absoluteError} = \sum_{j=1}^{36} \frac{|response_{model_j} - response_{data_j}|}{standardError_j} \quad (4.4)$$

$$Fitness_{groupError} = \sum_{j=1}^{36} |kMeans(response_{model_j}, 3) - kMeans(response_{data_j}, 3)| \quad (4.5)$$

Note that Duffy and Page showed each primate each stimulus 6-8 times throughout the experiment. The recorded firing rate value in the data set for each stimulus is the average of all the trials performed and has an associated standard error (i.e. standard deviation from the mean). The absolute error fitness calculation takes the standard error of each stimulus into account by using it to divide the difference between the model output and recorded data output. Thus, noisy data with high standard error values have less of an impact on the absolute error fitness summation. Adding the standard error to the average firing rate value gives us a maximum acceptable firing rate. Subtracting the standard error from the average firing rate error gives us a minimum acceptable firing rate. The acceptable range of firing rates values falls between this minimum and maximum.

Group error fitness is calculated to account for variations in neuron firing rate magnitude caused by noise in the recording process. K-means is applied to cluster responses into 3 classes, corresponding to low, medium, and high firing rates. Often the high firing rates correspond to excitatory responses and the low firing rates correspond to inhibitory responses. Due to the effect of training on noisy data, the DGM may produce segmental response values that differ in magnitude from the recorded response values but share the same general pattern of excitation and inhibition. Clustering the responses in this fashion reveals similarities in the response patterns. Unlike the absolute error fitness metric, exact firing rate values are not important. Once the classes for each segmental response are calculated by the k-means algorithm, the difference between the model class assignments and data class assignments are summed. For example, if a segment response produced by the model is assigned to the low firing rate class and the actual segment response recorded in the data was assigned to the high firing rate class, that segment would contribute a value of 2 to the overall group error for that stimulus.

### **4.3 Dual Gaussian Genetic Algorithm**

Yu's GA begins with a population of 2550 members. The parameters of each member are initialized randomly. Each of the 25 best performing members according to absolute error fitness are selected to form one set of parents and each of the 25 best performing members according to group error fitness are selected to form another set of parents. Each parent from each set

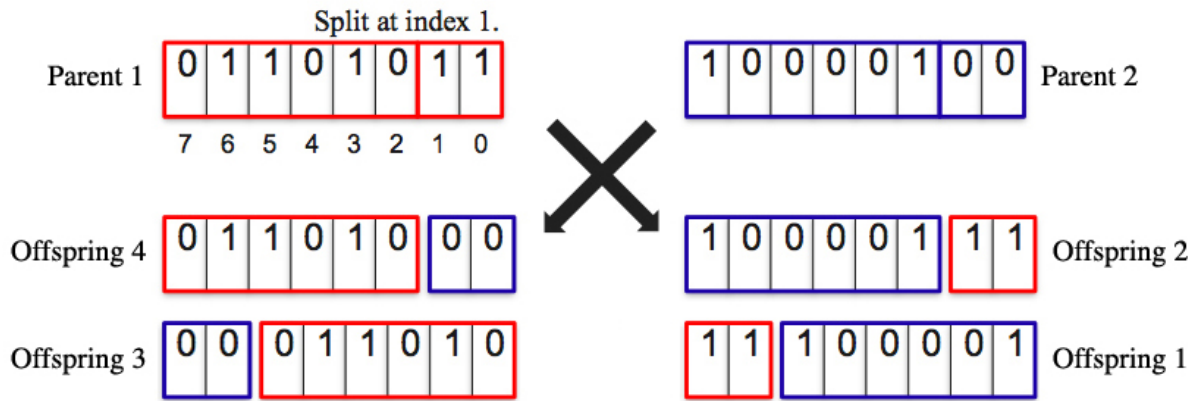


Figure 4.2: GA crossover example showing the generation of 4 offspring parameter values from 2 parent parameter values.

is then bred with each parent from the other set, which is a total of 625 parent pairings. Since each pair breeds 4 offspring via a crossover process, a total of 2500 offspring are produced. Reintroducing the 50 parents back into the population results in a population of 2550 members once again. Before the new population becomes the next generation there is a random chance that each of the 72 DGM parameters undergoes a form of mutation, which simply involves flipping a random bit in the binary representation of that parameter value. The selection, breeding, and mutation stages are repeated for each generation that the GA is run. After the last generation, the best performing member according to absolute error fitness is returned as the result. Parents breed by crossing over each of the 72 DGM parameters using the method shown in Figure 4.2. Note that each DGM parameter is mutated and crossed over independently of the other parameters.

Given each parent's parameter value in binary representation, the GA chooses a random index,  $i$ , corresponding to a bit location. Considering that the binary representation consists of  $n$  bits, the first offspring parameter value is formed by taking bits 0 to  $i$  from the first parent and appending bits  $i + 1$  to  $n$  from the second parent. The second offspring parameter value is formed by taking bits  $i + 1$  to  $n$  from the second parent and appending bits 0 to  $i$  from the first parent. The third offspring parameter value is formed by taking bits 0 to  $i$  from the second parent and appending bits  $i + 1$  to  $n$  from the first parent. Finally, the fourth offspring parameter value is formed by taking bits  $i + 1$  to  $n$  from the first parent and appending bits 0 to  $i$  from the second parent. Each of the 4 offspring parameter values are checked to ensure that they fall within an acceptable range for the parameter in question. If any one of them falls outside of that range it is capped to a minimum or maximum value, thus ensuring that all offspring are well-formed.

Yu developed two variations of the DGM to more accurately predict flow stimuli responses by introducing the concept of gain factors. Do not confuse gain factors with the multiplicative constant ( $C$ ) applied to each Gaussian. The multiplicative constant is multiplied with a Gaussian probability and polarity factor to determine the output of the Gaussian. Gain factors are applied to the Gaussian output after that calculation has been performed. The first DGM variation assigns one gain factor ( $Gf$ ) to each of the 9 screen segments to account for segmental dominance. The gain factor is multiplied with the response output by the summation of the two Gaussians for that segment, as



shown in Equation 4.6.

$$response_{model_j} = baserate + \sum_{i=1}^9 (Gf_i * response_{segment_i}) \quad (4.6)$$

The second DGM variation assigns one gain factor to each of the two Gaussians for each segment to account for segmental dominance and directional shift. The gain factor is multiplied by each Gaussian output before the outputs are summed for each segment, as shown in Equation 4.7.

$$response_{model_j} = baserate + \sum_{i=1}^9 (Gf_{i_1} * C_{i_1} * G_{\mu_{i_1}, \sigma_{i_1}}(x) * P_{i_1} + Gf_{i_2} * C_{i_2} * G_{\mu_{i_2}, \sigma_{i_2}}(x) * P_{i_2}) \quad (4.7)$$

Since each Gaussian may be centered around a different mean direction, increasing the magnitude of the response of one of those Gaussians will cause the summed output of both Gaussians to shift in favor of the mean direction for the Gaussian whose magnitude was increased (depending on whether that Gaussian is inhibitory or excitatory). The gain factor values are determined by training on the flow data using the same GA approach used to train the other DGM parameters after those values have been trained on singles or doubles data. The  $(\mu, \sigma, C, P)$  parameters are not modified while training the gain factors.

## 4.4 Doubles Template Matching

In addition to using the standard DGM trained on singles data (and its variations employing gain factors), Yu developed another approach to predict flow stimuli responses by constructing templates from DGMs trained on doubles data. Each doubles template corresponds to one direction of motion within the X-configuration hotspot and one direction of motion within the O-configuration hotspot, which comes to a total of  $4 * 4 = 16$  templates for each of the possible combinations of cardinal motion within both hotspots. Each doubles template consists of 9 segments corresponding to the 9 screen segments, and each segment (excluding the 2 hotspot segments) is modeled using 2 overlapping Gaussians, which are trained using the GA in Section 4.3.

The major difference between the doubles template and standard DGM is that the doubles template Gaussians are trained using doubles data as opposed to singles data. This is accomplished by training a separate DGM for each pairing of hotspot and non-hotspot segments. The X-configuration hotspot is paired with each of the 4 X-configuration non-hotspots and the O-configuration hotspot is paired with each of the 3 O-configuration non-hotspots. Thus, 7 DGMs are trained per doubles template and each DGM is trained on the 4 data points corresponding to the 4 cardinal motions within the non-hotspot. Although each doubles stimulus consists of hotspot and non-hotspot segment motion, the hotspot segment motion is encoded in the DGM because it is the same for all of the data points used for training that DGM. Therefore, the hotspot segment motion does not need to be taken into further

consideration when training the DGM. We are left with non-hotspot segment motion, which takes the same form as singles stimuli. Therefore, training each DGM on a subset of doubles data set is the same as training a DGM on a singles data set. The resulting doubles template is used to predict the flow stimulus response using Equation 4.3, which simply sums the individual segment responses. Template construction is illustrated in Figure 4.3.

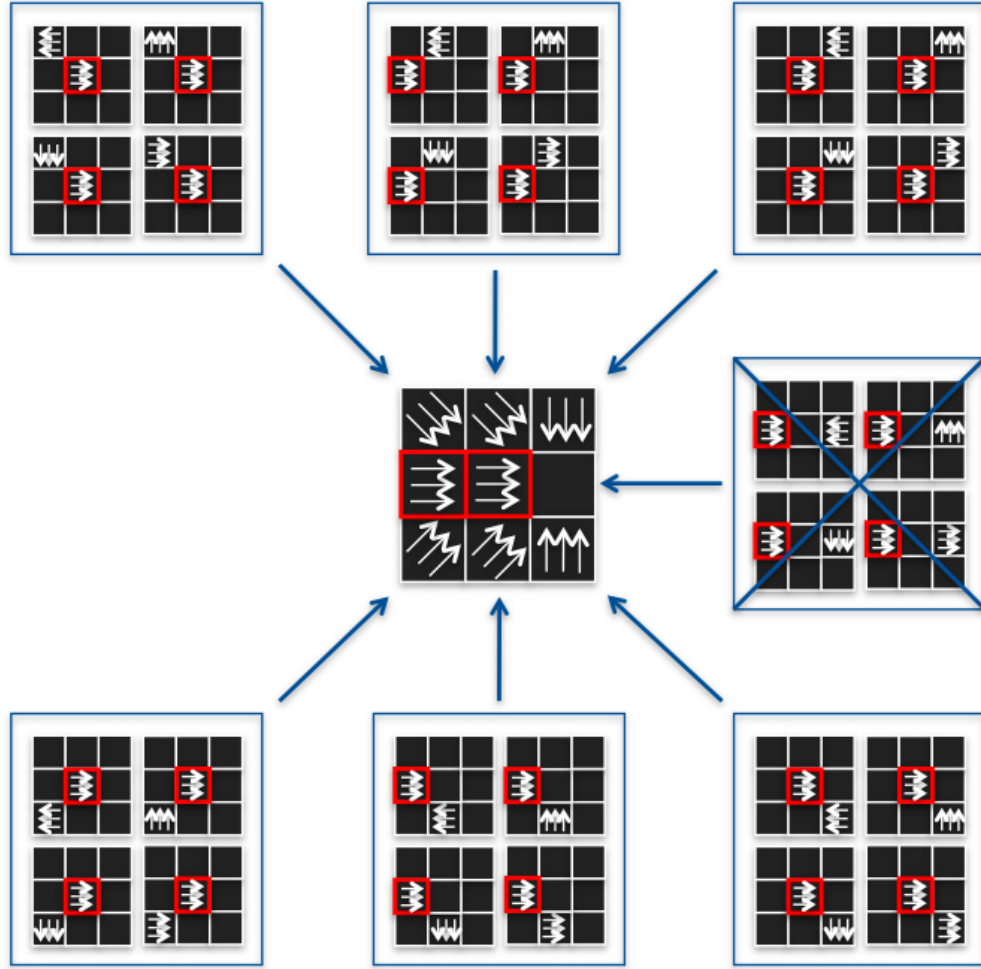


Figure 4.3: Example of doubles template construction. Hotspots are indicated by a red border. We wish to predict the firing rate response for the flow stimulus shown in the center plot. A DGM is trained for each pairing of hotspot and non-hotspot segments. Each of the 7 groupings of outer plots show the 4 doubles stimuli used to train each DGM corresponding to a non-hotspot segment (indicated by the blue arrow). Although a DGM is trained for the center rightmost non-hotspot segment, it contributes a value of 0 to the flow stimulus response because there is no motion within that flow stimulus segment. The 6 other DGMs are used to calculate a response according to the direction of motion within their corresponding non-hotspot segments. The response contributions are then summed to calculate the final prediction value. Images inspired by those in [17] and [23].

# Chapter 5

## Methodology

### 5.1 Scaled Gaussian Mixture Model Approach

Using a GA approach, Yu's DGM can reasonably predict flow stimulus responses when trained on singles or doubles stimulus responses; however, it does not accurately reflect the physiological structure of neuron RFs within the visual system of the mammalian brain. Specifically, Yu's DGM is structured based on an assumption that arose as part of Duffy and Page's experimental procedure. The visual stimuli shown to the primate was composed of groups of moving white dots. Each group occupied one of the 9 conceptual screen segments shown in Figure 3.1(b) and in turn each group occupied part of the primate's visual field. The boundaries between the segments were indistinguishable to the primate such each flow stimulus appeared as one continuous motion composed of dots of uniform size, density, and speed covering the primate's entire visual field. Yu's DGM is based on the false notion that the RF for each of the primate's MSTd neurons is composed of 9 regions corresponding to the 9 screen segments. We refer to this issue as the 9 segment constraint.

In order to do away with the structural restriction that the 9 segment constraint imposes upon the formation of the DGM, we extend Yu's work

by considering that a MSTd neuron RF can be modeled by a composition of Gaussian distributions of arbitrary location within the RF space. Like the DGM, each Gaussian is assigned a mean, polarity factor, multiplicative constant, and standard deviation. This composition of Gaussians forms a scaled Gaussian mixture model (SGMM). The traditional Gaussian mixture model (GMM) is shown in Equation 5.1.

$$q(x) = \sum_{i=1}^{numg} w_i p(x, \mu_i, \sigma_i) \quad (5.1)$$

The contribution of each Gaussian is calculated using the probability distribution function (PDF) ( $p(\cdot)$ ) according to the observed value ( $x$ ), mean ( $\mu$ ), and standard deviation ( $\sigma$ ) along the directional axis for that Gaussian, which is then multiplied by a weight ( $w$ ). The contributions are then summed to form the final firing rate response for the visual stimulus being observed, which may cover the entire visual field and overlap multiple Gaussians in the RF space. The sum of the weights assigned to each Gaussian in the mix equals exactly 1 [8].

The SGMM we employ is slightly different from the traditional GMM in that the multiplicative constants (i.e. weights) assigned to each of the Gaussians in the RF feature space are not constrained so that they must sum to 1. Instead the parameters of the Gaussians are trained in such a way that the summation of firing rate response contributions from each Gaussian equals the firing rate response value for the visual stimulus being observed. Therefore, the response output associated with each Gaussian not only varies

in accordance with the other Gaussians in the RF feature space, but also in accordance with the multiplicative constants assigned to each Gaussian. If the weights assigned to each Gaussian in the traditional GMM are thought of as percentages that must sum to 100%, the multiplicative constants assigned to each Gaussian in the SGMM can be thought of as scaling factors which ensure that the combined outputs of all of the Gaussians sum to 100% of the firing rate response for the visual stimulus being observed. Note that a multiplicative constant can be assigned a negative value, so this analogy is not entirely accurate.

## 5.2 Multivariate Gaussian Distributions

The primary difference between the DGM and SGMM is the RF feature space used by each model. The DGM RF feature space is 1-dimensional because the Gaussian distributions formed vary in terms of their directional preferences. The relative  $(x, y)$  spacial location of the Gaussian distributions are determined by which one of the 9 conceptual screen segments they occupy. Therefore, the spacial locations of the 2 Gaussian distributions assigned to each segment by the DGM are unalterable. The SGMM RF feature space is 3-dimensional because the  $(x, y)$  location of the Gaussian distributions can be freely varied in addition to the Gaussian distribution's directional preference. Thus, the SGMM is composed of multivariate Gaussian distributions in the  $(x, y, \text{dir.})$  RF space. The shape, size, and alignment of each Gaussian is determined using a covariance matrix and the SGMM produced has no notion

of screen segments. To associate a stimulus shown in the projector screen to the (x, y) plane of the RF space, each of the conceptual 9 screen segments is translated to relative coordinates in the (x, y) plane.

The 3-D Gaussian distributions used in the SGMM are more complex than the 1-D Gaussian distributions used in the DGM. Recall the equation used to represent a 1-D normalized Gaussian distribution along the directional axis:

$$G_{\mu,\sigma}(x) = e^{\frac{(x-\mu)^2}{2\sigma^2}} \quad (5.2)$$

Note that  $G_{\mu,\sigma}(x)$  is simply a parameterized form of the PDF  $p(x, \mu_i, \sigma_i)$  shown Equation 5.1. Similarly, equation 5.3 is used to represent a normalized 3-D multivariate Gaussian in (x, y, dir.) space and is the parameterized form of the multivariate PDF.

$$G_{\mathbf{M},\mathbf{\Sigma}}(\mathbf{x}) = e^{-0.5(\mathbf{x}-\mathbf{M})^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mathbf{M})} \quad (5.3)$$

Note that  $\mu$  in Equation 5.2 is the mean direction along which the Gaussian is most responsive while  $\mathbf{M}$  in Equation 5.3 is a vector consisting of an x coordinate, y coordinate, and direction. The direction retains the same meaning and the (x, y) coordinates denote a segment location in Cartesian space, as shown in Figure 5.1(a). Also,  $\sigma$  in Equation 5.2 is standard deviation while  $\mathbf{\Sigma}$  in Equation 5.3 is a covariance matrix of the form:

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xd} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yd} \\ \sigma_{dx} & \sigma_{dy} & \sigma_{dd} \end{bmatrix} \quad (5.4)$$



Recall the equation used to calculate the response contribution of a 1-D Gaussian where the  $(x, y)$  position is encoded in the location of segment  $i$ :

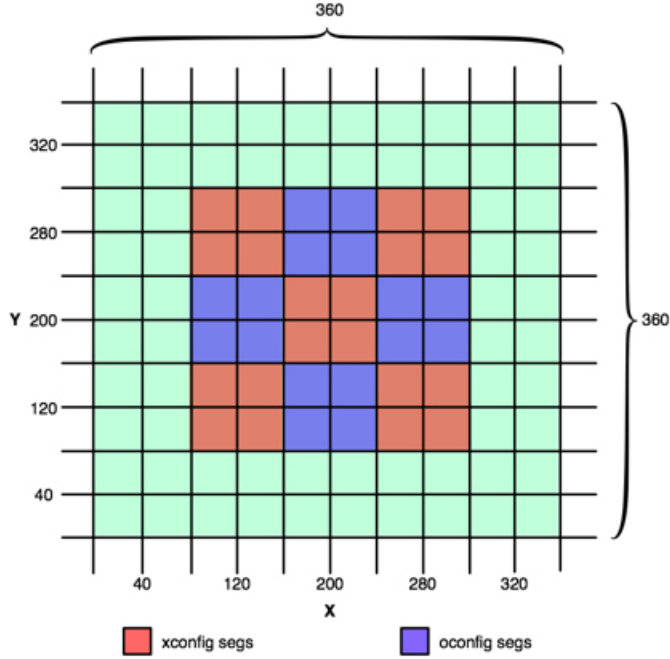
$$response_{segment_i}(x) = C_i * G_{\mu_i, \sigma_i}(x) * P_i \quad (5.5)$$

The same form can be used to calculate the response contribution of a 3-D Gaussian:

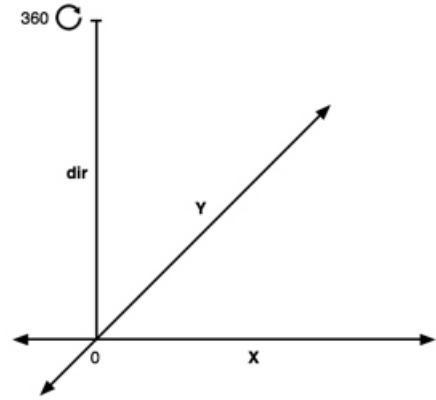
$$response(\mathbf{x}) = C * G_{\mathbf{M}, \mathbf{\Sigma}}(\mathbf{x}) * P \quad (5.6)$$

Note that the segment location is now encoded in the  $(x, y)$  elements of input vector  $\mathbf{x}$  along with the stimulus direction for that segment. Multivariate Gaussians of this form appear as ellipsoids in 3-D space, as shown in Figure 5.2. Note that another difference between the DGM and the SGMM is that the latter can represent Gaussians that extend outside of the RF space in which visual stimuli were shown because each multivariate Gaussian is not constrained to locations within the 9 segment screen space.

Multiple Gaussian distributions can occupy the same RF space, as shown in Figure 5.3. Similar to summing together the responses of the 2 Gaussian distributions for stimulus motion within a segment using the DGM, the response of overlapping Gaussian distributions for a given input vector in the SGMM are summed to determine the final response. The multivariate Gaussian concept is strongly based on Duffy and Wurtz's overlapping gradients hypothesis [12], as discussed in Section 3.5.

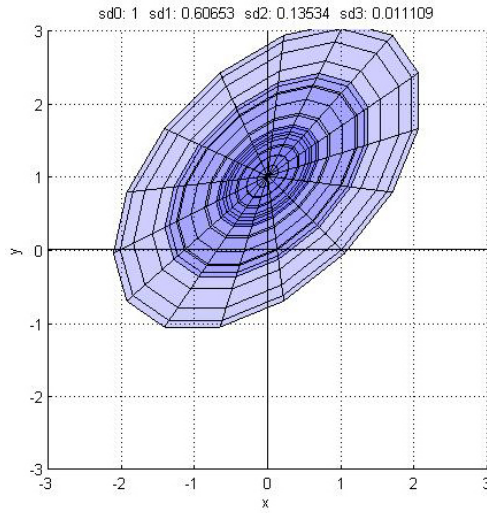


(a) Cartesian space representation.

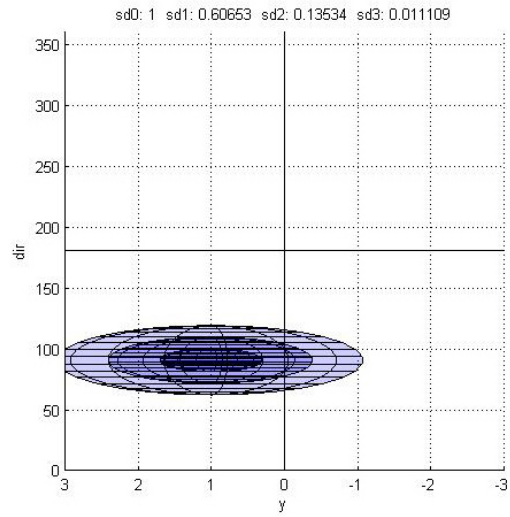


(b) 3-D RF feature space.

Figure 5.1: Multivariate Gaussian space representation. (a) Each of the 9 segments occupies a discrete area in the  $(x, y)$  plane. X-axis and Y-axis units do not correlate to actual sizes or positions of RFs or visual stimuli. An arbitrary scale was chosen to allow for relative sizing and positioning of Gaussians by the model. Notice the extra space that surrounds the X-configuration and O-configuration segments which allow for Gaussian placement near the boundaries of the RF. (b) 3-D RF  $(x, y, \text{dir})$  space. Note the directional rollover at the  $360^\circ$  position back to the  $0^\circ$  position.

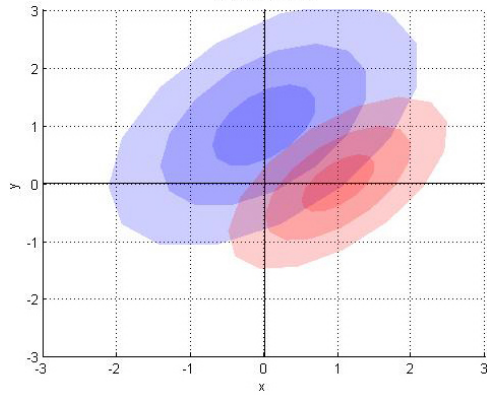


(a)  $(x, y)$  plane.

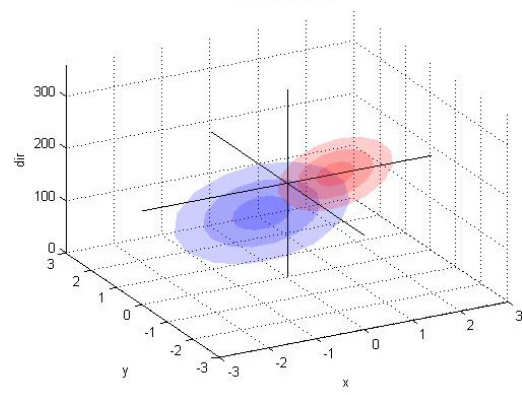


(b)  $(y, \text{dir.})$  plane.

Figure 5.2: Multivariate Gaussian distribution in  $(x, y, \text{dir.})$  space. Concentric ellipsoids represent standard deviations. The further away from the center of the ellipsoid, the lower the response magnitude.



(a)  $(x, y)$  plane.



(b)  $(x, y, \text{dir.})$  space.

Figure 5.3: Two multivariate Gaussian distributions sharing the same space. The red Gaussian is excitatory while the blue Gaussian is inhibitory.

### 5.3 Expectation Maximization Consideration

Originally we considered using traditional statistical methods such as Expectation Maximization (EM) to identify clusters of data points and generate a Gaussian model capable of predicting firing rate response values for various visual stimuli. First let's discuss the representation of stimuli properties in RF feature space. Then let's discuss a naive but optimal method for modeling the firing rate response values produced by singles stimuli, why that method is undesirable, and how it is related to EM.

When the primate experiments were performed, doubles and singles stimuli were shown within one or two of the 9 conceptual segments, respectively. Since we discretize the fluid motion associated with each flow stimulus into these 9 segments, the observed  $(x, y)$  position for stimulus motion within each segment is equivalent to the center of one of 9 corresponding  $(x, y)$  positions in RF feature space, as shown in Figure 5.1(a). Singles and doubles experiments were performed such that the motion within any one of the segments corresponds to one of the four cardinal directions. Therefore, the observed directional value for those visual stimuli always equals one of four corresponding directional positions in the RF feature space. All in all, there are a total of  $9 * 4 = 36$  uniformly distributed  $(x, y, \text{dir.})$  positions in the RF feature space that we are concerned with when modeling the singles data. The singles data model forms the foundation for modeling the doubles and flow data, which we will discuss later.

Note that each of the recorded singles data points corresponds to exactly one of the 36 possible (x, y, dir.) positions in the RF feature space. If we were to construct 36 corresponding non-overlapping Gaussians such that each of their means equals one of those positions, it is possible to create a SGMM with zero absolute error simply by assigning each Gaussian a multiplicative constant that equals the firing rate of the Gaussian’s corresponding singles data point firing rate when multiplied by the Gaussian’s probability. The shape, size, and alignment of the Gaussian is irrelevant as long as it does not extend beyond its corresponding segment borders. In effect, each Gaussian can occupy an infinitesimally small space at the center point of the segment it occupies because as long as the multiplicative constant for each Gaussian is scaled appropriately, the SGMM will have zero absolute error. Clearly such a model is overtrained and not useful if we wish to predict the firing rate for a singles stimulus that is not aligned with one of the 9 conceptual segments. Although we do not possess data for such stimuli, our model should be constructed with such considerations in mind. For example, what if we wish to predict the firing rate for a singles stimulus which occupies a portion of the primate’s visual field that is equidistant between two of the 9 conceptual segments? The overtrained model will fail because there would be no Gaussian that overlaps that position in the model. We refer to this as the one to one Gaussian to singles data point correspondence issue.

A SGMM with non-trivial predictive capabilities should be composed of Gaussians with relevant shapes and sizes that occupy non-trivial RF feature

space. In other words, we wish to “fatten” the Gaussian distributions which have a tendency to shrink around segment centers. Also, we do not want to limit ourselves to forming Gaussians based on the uniformly distributed nature inherent in the segmental positions of the singles stimuli. Many traditional statistical methods are not capable of addressing these concerns without significant modification. Specifically, EM is not a desirable choice for our modeling problem because:

1. By maximizing the estimated probability of the sample mean, EM has a propensity to generate Gaussians with minimal overlap, in effect forming data clusters [6]. As discussed in Section 3.5, Duffy and Wurtz’s overlapping gradient’s hypothesis encourages the use of overlapping Gaussians in the model in order to exhibit the effects of various excitatory and inhibitory RFs that contribute to the overall firing rate response for the stimulus observed.
2. In order to employ EM, an additional dimension would need to be added to the (x, y, dir.) RF feature space to account for variation in firing rate magnitude. Considering that there are only 36 singles data points recorded in that 4-D space, attempts to employ EM to such a data space would be plagued by the curse of dimensionality.
3. Three of the four features in that 4-D space are specified as a result of experimental procedure and therefore are not suitable for clustering purposes on their own: the x position, the y position, and the motion

direction. The firing rate should be a primary consideration and treated differently from the other features.

4. There is no simple way to constrain the shape, size, and alignment of Gaussians formed by EM without making non-trivial modifications to the operation of the algorithm. The Gaussians formed by EM may be too small for our purposes, be badly scaled, or have a singular covariance matrix (i.e. no inverse), making calculation of the PDF problematic.
5. We represent Gaussian directional preference as one of the RF feature space dimensions along a Cartesian axis. Standard algorithmic techniques do not consider the directional overlap that occurs between  $0^\circ$  and  $360^\circ$ .

Due to the aforementioned issues associated with employing EM, we decided to create our own algorithm for training SGMMs. As part of the process, we re-evaluated the use of the PDF employed by the traditional GMM, as we will discuss in the next section. Note that some of the EM issues discussed could have been addressed by using regularization terms in the objective function to ensure the generation of Gaussians of a certain size and other properties.

## 5.4 Probability Function and Directional Covariation Considerations

If we were to use the traditional GMM we would multiply the PDF output of each multivariate Gaussian by a multiplicative constant and polarity factor and sum each of those Gaussian contributions together to calculate the firing rate response for a visual stimulus. One of our primary concerns with using the PDF is that it only calculates a probability at a specific point in space. When modeling singles data, those points correspond to the centers of each of the 9 conceptual segments at the four cardinal directions. As far as the PDF is concerned, the space in between the segment centers is irrelevant because no singles stimuli are aligned with that space; however, to increase model generality so that we are reasonably able to predict firing rate responses for singles stimuli that do not correspond to one of the 36 singles stimuli RF space positions in the training set, the space in between segment centers is relevant.

This issue is related to creating a model capable of predicting firing rate responses for singles stimuli of various sizes. A model employing the PDF is only capable of considering the center point of a stimulus as it maps to a  $(x, y)$  position in RF space and therefore is incapable of taking into consideration the size of the stimulus. Although all 36 of the singles stimuli recorded by Duffy and Page are of equal size and are aligned with exactly one of the 9 conceptual segments, the concern is still valid and is worth our consideration in order to develop a more robust model. One way to take the



stimulus size into consideration is to perform integration across the stimulus area as it maps to the  $(x, y)$  plane in RF space. The cumulative distribution function (CDF) becomes a natural choice because it can be computed as the integral of the PDF.

Note that the multivariate Gaussian CDF does not possess a nice closed form like the multivariate PDF and thus requires Gaussian quadrature approximation (GQA) to calculate. Essentially, GQA is a technique that subdivides the integration space and estimates the volume within each space while reducing error to within an acceptable threshold [1]. Note that using GQA to calculate a multivariate CDF is a quite computationally intensive compared to calculating a multivariate PDF. Although Matlab was used to implement and run most of the model training procedures and data analysis discussed in this thesis, it was necessary to implement GQA in C++ code and integrate it into Matlab using the MEX interface in order to run our programs within a reasonable amount of time. Even with this optimization it still took approximately 3 hours to generate and validate one CDF SGMM with 10 Gaussians on a 2.4 GHz Intel Core Duo MacBook Pro laptop with 4 GB of RAM. Comparatively, it took less than 20 minutes to generate and validate one PDF SGMM with 10 Gaussians on the same machine.

Employing the CDF does not change the overall form of the characteristic SGMM equation, as shown in Equation 5.1. The CDF simply replaces the PDF. Also, unlike the PDF which operates on a single point in the RF feature space, the CDF integrates over a region of  $(x, y)$  points corresponding

to a specific direction in that space; however, since integration in 3-D space requires a volume and not a plane, a region of  $(x, y)$  points is not enough. Thus, the CDF is calculated over a  $+5/-5^\circ$  directional range to form a volume of  $(x, y, \text{dir.})$  space. Integrating over a directional range is also beneficial because it also allows the model to generalize better.

Both the PDF and CDF calculations depend on the shape, size, and alignment of the Gaussian in question, which raises an important question: Should we allow Gaussian distribution directional preference to vary with  $(x, y)$  position? We refer to this behavior as directional covariation. A covariance matrix which does not allow for directional covariation is shown in Equation 5.7a and a covariance matrix which allows for directional covariation is shown in Equation 5.7b.

$$\Sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & 0 \\ \sigma_{yx} & \sigma_{yy} & 0 \\ 0 & 0 & \sigma_{dd} \end{bmatrix} \quad (5.7a)$$

$$\Sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xd} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yd} \\ \sigma_{dx} & \sigma_{dy} & \sigma_{dd} \end{bmatrix} \quad (5.7b)$$

Without directional covariation Gaussians in the  $(x, y, \text{dir.})$  space are constrained to being perpendicular to the directional axis, as shown in Figure 5.4(a). No directional covariation assumes independence between the  $(x, y)$  position within a Gaussian and its directional preference. Gaussians of this form are in accordance with the gradients described in the overlapping gradient hypothesis, as discussed in Section 3.5, and are primarily responsive to one direction. On the other hand, Gaussians with directional covariation are

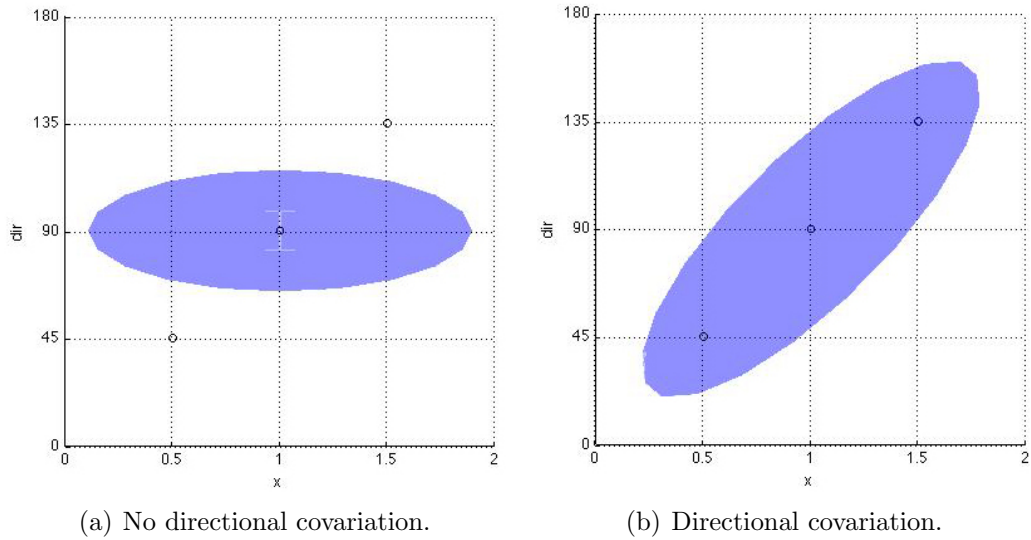


Figure 5.4: Examples of Gaussians with and without directional covariance. It would take three Gaussians of the size and shape shown in (a) to overlap the three points shown while it only takes one Gaussian with directional covariance, as shown in (b).

not as constrained and may be responsive to a range of directions because of a dependency between the  $(x, y)$  position within a Gaussian and its directional preference, as shown in Figure 5.4(b). The major benefit of directional covariance is that it allows for more variability per Gaussian and therefore fewer Gaussians may need to be trained to fit the singles data. Fewer Gaussians may help prevent over-fitting because fewer Gaussians substantiates moving away from the one to one Gaussian to singles data point correspondence issue mentioned in Section 5.3.

## 5.5 Training in Stages

Like the DGM, we employ a GA to train the SGMM Gaussian parameters; however, unlike the DGM, our GA employs tournament-style selection and trains the SGMM in stages. Tournament-style selection is performed by randomly partitioning the population into sets of 4 members. The 2 best performing members of each set in terms of lowest squared absolute error are selected to be the parents. The parents then breed 2 offspring (offspring 2 and offspring 4 in Figure 4.2) to replace the 2 worst performing members of the same set. The 2 parents and 2 new offspring are then carried over into the population for the next generation of training.

Each stage of the GA training process trains one new Gaussian to add to the SGMM. By focusing on training one Gaussian on a time the GA considers only local information for each stage of training. For this reason training in stages can be considered a greedy approach, but we will introduce a greedier algorithm in Section 5.7. Initially each member of the GA population contains one Gaussian with randomly generated parameters. After a certain number of user-defined generations, the member of the population which produces the lowest squared absolute error is selected. In the next stage of training the member selected in the previous stage becomes a prototype from which new members are formed. In other words, each new member inherits the Gaussian trained in the previous stage and takes on a new Gaussian with randomly generated parameters. Only the parameters of the new Gaussian are modified by training performed in the current stage. Gaussians inherited from previous

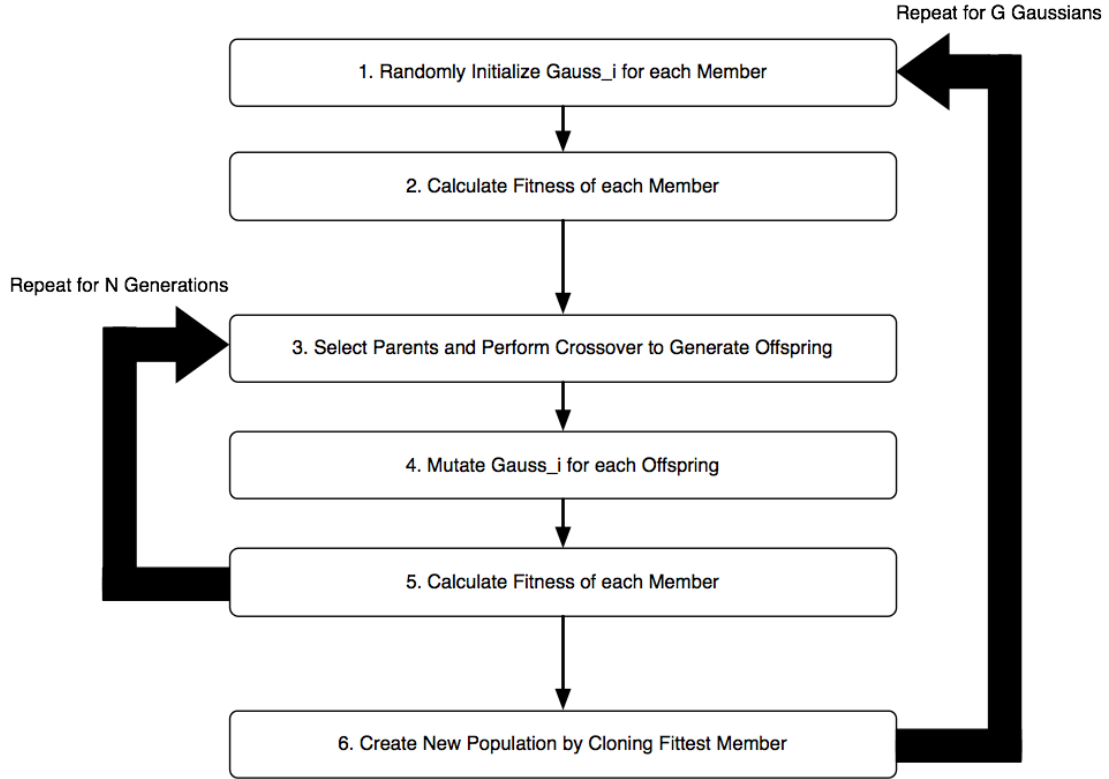


Figure 5.5: The staged training procedure behaves much like the GA used to train the DGM. The major difference is that each stage focuses on training the parameters of one new Gaussian instead of all of the Gaussians in the model. The final step of each stage clones the best member of the population so that each member of the next stage's population inherits all of the Gaussians trained in previous stages. A randomly generated Gaussian is then added to each member in the first step of the next stage.

generations are excluded from further modification and are directly passed on to the final SGMM. Training in stages continues until a specified number of Gaussians have been trained or until the absolute error for the SGMM is below a specified threshold. The staged training procedure is shown in Figure 5.5.

## 5.6 Training via Speciation

Each Gaussian in the SGMM covers a different volume of the RF feature space and has a different level of excitation or inhibition associated with it. The purpose of each Gaussian is to model the firing rates of one or more single data points corresponding to single stimuli responses within that volume. By addressing a subset of the data points, each Gaussian essentially solves a subproblem related to the overall problem of modeling all of the data points. During any given generation of GA staged training, each member of the population is attempting to solve a subproblem of this form and any two members may be attempting to solve different subproblems involving different data points in the RF feature space. These subproblems may have different optimal solutions realized by Gaussians of different shapes, sizes, and positions.

A concern is that the GA may attempt to crossover the parameters of two Gaussians attempting to solve different subproblems. This is undesirable because each subproblem may require Gaussians with vastly different parameter values. Breeding two vastly different Gaussians will often produce a child that does not excel at solving the subproblem being addressed by either parent. Thus, the offspring resulting from the crossover of two such Gaussians does not take full advantage of traits it inherits from its parents and serves little value other than introducing novelty into the model. In order to take full advantage of the traits an offspring inherits from its parents, each offspring should be bred from parents trying to solve the same subproblem. By dividing the GA population into species such that all of the members within each

species are trying to solve the same subproblem, we can take full advantage of the crossover operation by only breeding members within the same species. We refer to this as training via speciation.

The major benefit of training via speciation is that two Gaussians attempting to solve different subproblems do not compete against each other for survival or breeding rights. During each generation of training by a non-speciated GA the performance of every member within the population is evaluated and only the fittest members are selected to breed the next generation. In other words, all members of the population compete for the right to become parents and pass on their traits. On the other hand, during each generation of training by a speciated GA the performance of every member within each species is evaluated and only the fittest members within each species are selected to exercise those rights. Therefore, only members within the same species are allowed to compete against each other. Breeding rights are allocated to each species based on their fitness compared to that of other species and the best performing members from within each species are selected to breed with each other. Thus, species which are most likely to generate the fittest member of the population are given the most opportunities to do so.

Due to the randomized nature of the GA, it is very difficult to predict how well a given member of the population will perform in the future after undergoing training. Therefore, we want to protect innovation by not prematurely throwing out members which initially perform poorly but may perform well in the future after a few more generations of training. To protect innova-

tion we divide the population into species based on their similarity and only the members of each species compete amongst themselves for survival. The concept of speciation is based on the work of Ken Stanley and the NeuroEvolution of Augmenting Topologies (NEAT) methodology [14]. The speciated training process is shown in Figure 5.6.

Breeding rights are calculated such that the fittest species are allowed to breed more offspring than species that are not as fit. Equation 5.8 calculates the number of offspring that each species is allowed to breed [14].

$$N'_j = \frac{\sum_{i=1}^{N_j} f'_{ij}}{\bar{f}} \quad (5.8)$$

$N'_j$  is the new number of offspring species  $j$  is allowed to breed,  $N_j$  is the old number of members in species  $j$ ,  $f'_{ij}$  is the adjusted fitness of member  $i$  in species  $j$ , and  $\bar{f}$  is the mean adjusted fitness of the entire population. Equation 5.9 calculates the adjusted fitness [14].

$$f'_i = \frac{f_i}{\sum_{k=1}^N \partial(i, k)} \quad (5.9)$$

$f'_i$  is the adjusted fitness for member  $i$ ,  $f_i$  is the non-adjusted fitness for member  $i$ ,  $N$  is the size of the population, and  $\partial(i, k)$  is the compatibility between member  $i$  and member  $k$ . The compatibility between two members is calculated as the percentage of volume in the RF feature space shared by the Gaussians of those two members. Since the calculation of the intersecting volume between two ellipsoids is non-trivial, we approximate the percentage



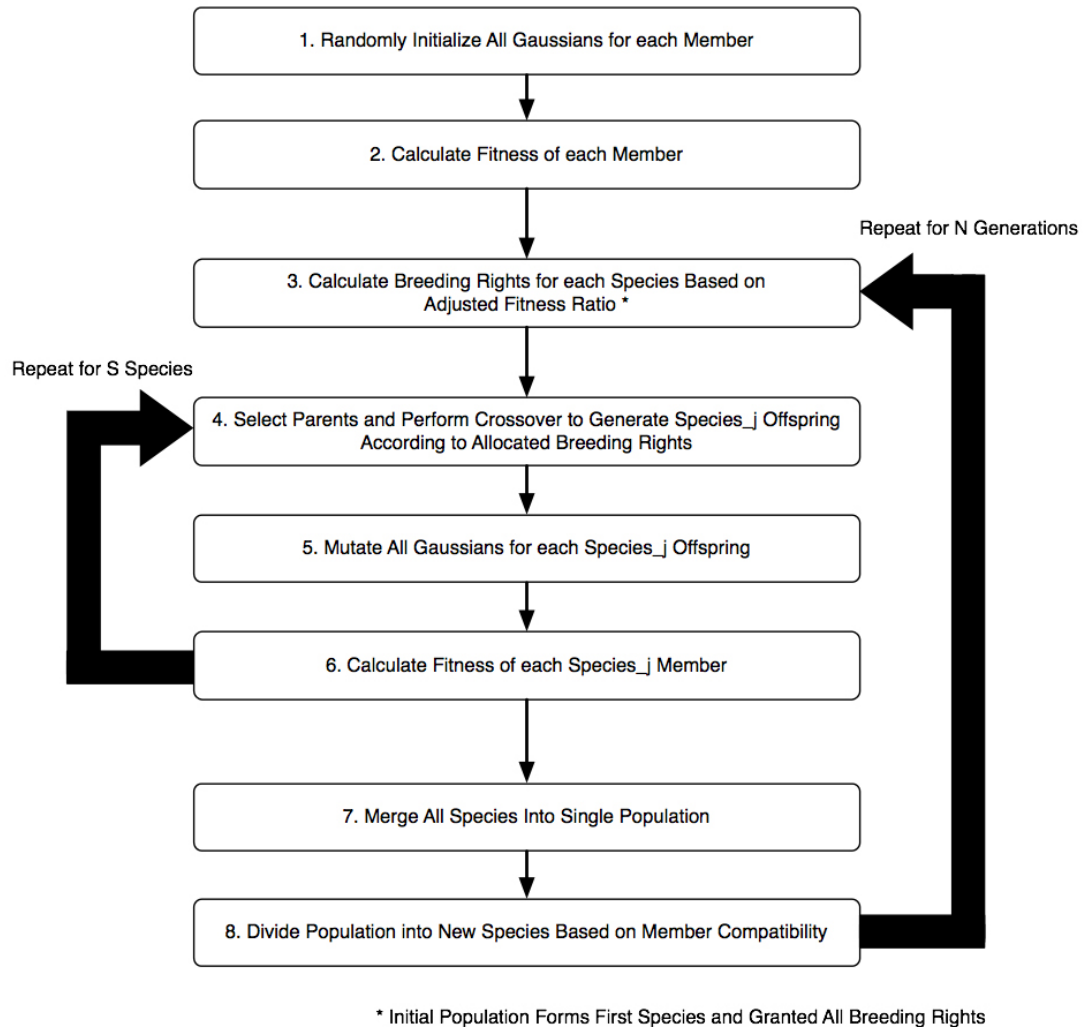


Figure 5.6: The speciated GA behaves much like the non-speciated GA. The major difference is that the population is divided into subpopulations, called species, so that the most compatible members are grouped together. Crossover is performed only between members of the same species. At the end of each generation of training all of the species with their new offspring are merged together into a single population and divided into species once again. This regrouping and redision is necessary because a mutated member may be more compatible with a species different from that of its parents.

by using cuboid representations of the Gaussian ellipsoids. Equation 5.10 calculates the non-adjusted fitness for a given member ( $x$ ):

$$f(x) = 1000 * \frac{1}{1 + Fitness_{MSE}(x)} \quad (5.10)$$

The mean square error (MSE) fitness function,  $Fitness_{MSE}$  is calculated over the the 36 singles data points, as shown in Equation 5.11.

$$Fitness_{MSE} = \frac{1}{36} \sum_{j=1}^{36} (response_{model_j} - response_{data_j})^2 \quad (5.11)$$

Clearly, as a member's MSE fitness decreases the member's non-adjusted fitness increases proportionally. A non-adjusted fitness value of 1000 is ideal. The coefficient 1000 allows for a wide distribution of fitnesses for selection. Please do not confuse the adjusted fitness and non-adjusted fitness functions used to calculate breeding rights in speciated training with the absolute error, group error, and MSE fitness functions shown in Equation 4.4, 4.5, and 5.11, respectively. The adjusted fitness and non-adjusted fitness functions calculate high values for the best performing members. On the other hand, the absolute error, group error, and MSE fitness functions calculate low values for the best performing members because they calculate the error between the model and the recorded data.

In order to determine if a member of the population belongs to a given species, that member is compared to a random member of the first species using the compatibility function to produce  $\partial$ . That member is then compared to a

random member of each other species in the order in which those species were created. As soon as we determine that  $\partial < \partial_t$  for a given species, where  $\partial_t$  is a predetermined compatibility threshold, the member under determination is assigned to that species. We choose a target number of species and the value of  $\partial_t$  is adjusted to best sort the population into that number of species. Also, we ensure that a minimum number of members is assigned to each species. This approach to dividing the population into species is the same as that performed by the NEAT methodology.

## 5.7 Greedy Training Algorithm

One of the inherent problems of all of the GAs discussed so far is granularity. Both crossover and mutation are performed on bit string representations of Gaussian parameter values. These values are integers and lack decimal precision, which is clearly beneficial when manipulating the values of multiplicative constants and covariances in continuous 3-D space. There are ways to convert decimal values into integer form for GA manipulation by multiplying each value by a known constant (often to the power of 10) and truncating trailing decimal places deemed unnecessary; however, this presents a problem because it increases the number of significant digits in the bit string representation. The more significant digits in the bit string representation, the larger the effective search space of the GA. The randomized nature of the crossover and mutation operations of the GA helps introduce novelty into the population being trained, but in doing so the GA fails to converge to specific

values in the search space (like a gradient descent method). Instead, the GA employs the fitness function to discard poorly performing members. There is no guarantee that a better performing member will be produced after each generation of the GA training.

Increasing the search space of the GA often increases the number of poorly performing members that it will produce because of its randomized nature. The more poorly performing members produced by the GA, the greater the number of generations of training required to produce a member of reasonable performance. The benefit of increasing the search space is that ultimately a member with precisely tuned parameters may be produced with better performance than a member produced by a GA operating within a more restricted search space. In practice, the number of generations of training required to produce a member of reasonable performance quickly becomes intractable as the precision required of the model (and associated search space) increases. Therefore, we limit the search space of our GA so that it completes within a reasonable number of generations and produces a model with reasonable performance.

When training the first few Gaussians in the model precision is not too important. The first few Gaussians provide a foundation upon which subsequent Gaussians can be tuned to address more specific regions of excitation and inhibition with the RF feature space. As the number of Gaussians in the model increases, the effectiveness of the GA often decreases because of the increased need for precision. To address this issue, we employ a greedy algo-

rithm (GRA). The GRA operates by exhaustively trying out each Gaussian from a set of Gaussians of predetermined size, shape, and position. The multiplicative constant for each Gaussian is precisely calculated in order to address the single data point with the most residual error for the current stage of training. The residual error calculation is shown in Equation 5.12 (standard error is discussed in Section 4.2). This algorithm is greedier than the staged training procedure because for each stage of training a specific data point is considered.

$$ResidualError(x) = \max\{0, |response_{model}(x) - response_{data}(x)| - standardError_{data}\} \quad (5.12)$$

Since both the probability of the Gaussian under consideration and the firing rate response value are known for that data point, the calculation of the multiplicative constant for the Gaussian simply becomes the division of the firing rate by the probability. The Gaussian which performs the best in concert with the existing Gaussians in the model, in terms of MSE fitness for all single data points, is selected and added to the model.

Clearly the GRA operates within an even more restricted search space than the GA because it is limited to selecting Gaussians from a predefined set. The benefit is that the GRA operates much quicker than the GA and ensures that Gaussians of certain shapes, sizes, and positions are considered. Also, the GRA is much more precise in its calculation of the multiplicative constant for each Gaussian than the GA. The GRA is greedy in the sense that it focuses on

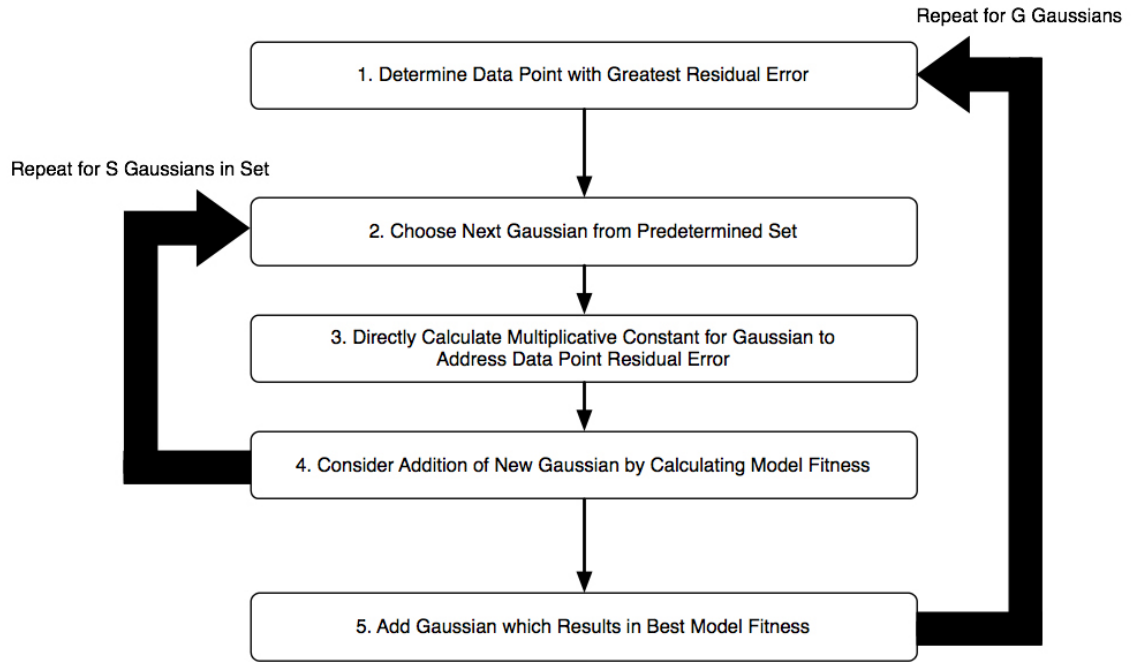


Figure 5.7: The greedy training procedure incrementally trains Gaussians like the GA staged trained procedure. First the data point with the greatest residual error is determined. Then each Gaussian from a predefined set is considered in turn to address the error. The Gaussian which results in the greatest fitness improvement for the model over all data points is added to the model.

individual data points to reduce the greatest residual error with the addition of each new Gaussian, thereby basing decisions on local information in the RF feature space, which leads to locally optimal choices; however, by considering those choices in respect to the MSE fitness over all data points we ensure that those choices are also globally optimal for the current stage of training. The GRA training procedure is shown in Figure 5.7.

## 5.8 Competitive Training Algorithm

In order to take advantage of the benefits of training in stages and training via speciation, we combine both approaches into a staged-speciated genetic algorithm (SSGA) which trains one additional Gaussian per stage by dividing that stage's population into species. After a predetermined number of generations of speciated training have passed for the current stage, the fittest member among all of the species is selected and passed on to the next stage of training. In turn the next stage focuses on training a new Gaussian that works in concert with all of the previously inherited Gaussians. No species information or members, other than the fittest member, are passed on to the next stage of training.

Compared to the GRA, the SSGA allows for a greater degree of freedom in the training process because it does not restrict the formation of Gaussians to predetermined sizes, shapes, and positions. Also, speciated training allows many subproblems to be considered at once and thereby protects innovation. On the other hand, compared to the SSGA, the GRA can produce much more precise results because of the way in which it directly calculates multiplicative constants and directly addresses residual error for specific data points.

To take advantage of the unique benefits presented by both the SSGA and GRA, we combine both algorithms into a competitive algorithm (CA). The operation of the CA is very simple. During any given stage of training, both the SSGA and GRA work independently to train a new Gaussian to add to the model. The Gaussian produced by each algorithm is then compared

against each other and the fittest one in terms of residual error is passed on to the next stage of SSGA training and the next stage of GRA training. The process repeats until a certain number of Gaussians have been added to the model or the residual error drops below a certain threshold.

## 5.9 Feature Space Considerations for Doubles and Flow Stimuli

We model singles data by using a SGMM trained by the CA. By extension, we can use a similar approach to model doubles data by expanding the  $(x, y, \text{dir.})$  space to 4 dimensions to account for the direction of motion within the hotspot segments. Since the X-configuration and O-configuration have disjoint hotspots, we would need to train a separate model for X-configuration doubles data and O-configuration doubles data. Note that the  $(x, y)$  position of the doubles hotspot for each configuration is not important because it is the same for all of the data points associated with that configuration. Only the non-hotspot segments change. Specifically, the RF feature space used for doubles stimuli would take the form  $(x, y, \text{dir.}, \text{hs dir.})$ . We can subdivide doubles data into separate RF feature spaces by employing 8  $(x, y, \text{dir.})$  spaces to account for each of the 4 cardinal directions of motion within each of the hotspot segments, which is similar to how Yu created 16 templates corresponding to the 16 possible combinations of cardinal motion within both of the hotspot segments, as discussed in Section 4.4. Yu used one template for both X-configuration and O-configuration data, while we are considering using



separate feature spaces for the X-configuration and O-configuration data.

We can take this approach one step further and consider modeling flow data by using a RF feature space that accounts for the directions of motion within both hotspot segments (which are both active simultaneously) as well as the directions of motion within each non-hotspot segment. Note that the  $(x, y)$  positions of the non-hotspots segments would not be important because the positions of those segments are the same for all of the data points and each segment is encoded in the feature space based on its ordinality. Specifically, the RF feature space used for flow stimuli would take the form (X-config. hs dir., O-config. hs dir., non-hs 1 dir., non-hs 2 dir.,  $\dots$ , non-hs 7 dir.). As mentioned when we considered employing EM in Section 5.3, there are not enough recorded data points to use a feature space greater than 3 dimensions without running into the curse of dimensionality. Thus, expanding the feature space for flow data is not a viable option due to the limitations on the amount of recorded data available, and the number of variable features between data points. As an alternative to Gaussian models, we considered traditional trend estimation techniques to predict flow stimuli firing rate response values.

## 5.10 Linear Regression Technique

Originally we considered constructing a neural network for predicting firing rate response values for doubles and flow stimuli, but we decided against the idea in order to develop a methodology which allows us to better understand the underlying model mechanics. Many non-trivial neural networks

operate in a convoluted fashion which makes it difficult to trace the calculation of output values back to the original input values. Oftentimes the reason why such networks are able to successfully recognize patterns in the training data can only be explained in general terms. In order to reduce the complexity of a neural network, one modification would be to use a pure linear transfer function to calculate neuron activation levels, thus reducing the operation of the neural network to a form of LR.

The key to modeling the non-linear behavior of neurons using a LR technique is extracting non-linear features from the data. Inspiration came from the work of Simard, et al., who developed a convolutional neural network to recognize handwritten characters in [19]. LeCun, et al., developed a similar network using a hyperbolic tangent transfer function for gradient descent backpropagation in [16]. Both networks consist of multiple layers of hidden neurons which extract feature information from a pixel array representation of a hand-written character to recognize. The structure and operation of neurons in these layers simulates the process of convolving randomly distributed convolutional kernels with the the pixel array. These kernels are referred to as feature maps and are trained by modifying the weights between hidden neurons. The combination of feature maps in the convolutional networks identify non-linear trends in the spacial distribution of hand-written characters and assign values to them. The final stages of the networks consist of fully-connected layers of neurons that classify characters based on the feature values calculated in the previous stages. Feature values are weighted, summed, passed through

an activation function, and compared to make the final classification.

Our modeling approach does not employ a neural network or convolution; however, like Simard and LeCun, we employ a feature extraction process in the initial stages of our technique to identify non-linear trends in the data (i.e. stimuli responses) and assign values to them. The final stages employ multiple LR, as described in [7] and [5], to assign weights to the feature values, which are then summed to calculate a firing rate response value. The spacial distribution of the doubles and flow stimuli is discretized into 9 segments. We calculate features according to the stimulus direction of motion within each of these segments to produce 9 sets of features. Motion was not shown outside of the 9 segments during the primate experiments, but if it was, such as between segments, we would also need to incorporate positional (x, y) features into each set. The features calculated within each segment depend on if the segment is a hotspot or non-hotspot, as we will discuss in the next section.

We employ multiple LR to solve Equation 5.13:

$$y = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_i x_i + c + \epsilon \quad (5.13)$$

Assuming there are  $i$  features, each  $x$  corresponds to a feature value and is multiplied by an associated weight  $\beta$ . These products are then summed with an optional  $c$ , a constant or intercept, and  $\epsilon$ , an error term used to account for noise, to calculate  $y$ , the neuron firing rate in spikes/second for the a given stimulus minus the base firing rate for the stimulus type (singles, doubles, or flow). LR focuses on the conditional probability distribution of  $y$  given the

values of each  $x$  and attempts to determine values for each  $\beta$  to minimize the sum of squares of the errors (residuals) when solving the equation. Specifically, we used the Matlab “regress” function to perform LR.

### 5.11 Feature Selection

The features provided as inputs to the LR technique focus on the directional selectivity of each neuron. As per our discussion of opponent organization in Section 3.4, we believe that the hotspot singles firing rate responses can be used to predict the firing rate responses for doubles and flow stimuli by determining which directions of motion the hotspot exhibits the greatest response. Thus, firing rate response magnitude and direction of motion are primary elements of the hotspot features provided as inputs to the LR technique, as listed in Table 5.1.

Hotspot Features
hotspot contrib.
sin hotspot dir. * hotspot contrib.
cos hotspot dir. * hotspot contrib.

Table 5.1: Hotspot segment features.

A segment’s firing rate response contribution is the firing rate response produced by the neuron for stimulus motion shown only within that segment. It is the sum of each of the Gaussian responses for the stimulus and segment in question. The segment firing rate response contribution for a singles stimulus equals the firing rate response for the entire stimulus because only one segment

is active in a single stimulus. By extension, each double stimulus has 2 segment response contributions and each flow stimulus has 8 or 9 contributions.

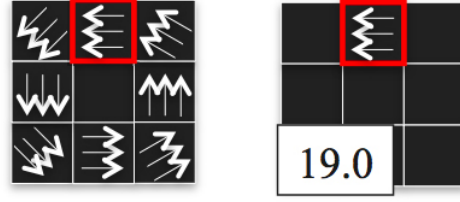
We enforce a relationship between the hotspot's contribution and the horizontal and vertical components of the direction of motion within the hotspot segment, respectively. Since we consider rightward motion to be  $0^\circ$  from the directional axis, the cosine of the direction of motion within the segment is its horizontal vector component ( $\cos(0^\circ) = 1$ ), and the sine of the direction of motion within the segment is its vertical vector component ( $\sin(90^\circ) = 1$ ). By multiplying the hotspot's contribution by its horizontal vector component we scale the segment contribution before the LR process assigns it a weight. We call this pre-weighting and it is desirable because we assume that the hotspot has a preference for either  $0^\circ$  or  $180^\circ$  horizontal motion, again as mentioned in our discussion of opponent organization. Thus, the hotspot contribution will be weighted by 1 for  $0^\circ$  horizontal motion or by -1 for  $180^\circ$  horizontal motion, due to the effect of the cosine multiplication. This is acceptable if the hotspot has a preference for  $0^\circ$  motion because  $180^\circ$  motion is penalized by the -1 weight, but it is not acceptable if the hotspot has a preference for  $180^\circ$  motion. In that case it is exactly the opposite of what we want. In such situations we rely on the weight assigned by the LR process to reverse the sign of the hotspot horizontal contribution feature. Similarly, we assume that the hotspot has a preference for either  $90^\circ$  or  $270^\circ$  vertical motion, so we pre-weight the hotspot contribution by 1 for  $90^\circ$  motion and by -1 for  $270^\circ$  motion.

Pre-weighting the hotspot contributions using trigonometric functions enforces non-linearity in the input features provided to the LR process. Note that horizontal directions of motion within the hotspot will produce hotspot horizontal contribution features with higher magnitudes than those for diagonal and vertical motions (for example, because  $abs(cos(0^\circ)) > abs(cos(45^\circ)) > abs(cos(90^\circ))$ ). Similarly, vertical directions of motion within the hotspot will produce hotspot vertical contribution features with the highest magnitudes. Thus, cardinal directions of motion within the hotspot will always be weighted higher than non-cardinal directions of motion. This is desirable because we assume that the directional selectivity of neurons is most strongly aligned with the cardinal directions and tapers off as the direction of motion approaches intermediate diagonals. An example showing how hotspot features are calculated is shown in Figure 5.8.

Non-Hotspot Features
seg. contrib.
sin seg, dir. * seg. contrib.
cos seg. dir. * seg. contrib.
doubles pairing response
doubles pairing response - seg. contrib.

Table 5.2: Non-hotspot segment features.

The first three non-hotspot segment features shown in Table 5.2 are very similar to the three hotspot features we just discussed. The last two features shown in the table are based on information obtained from the recorded doubles data. All of the other features we have discussed so far are based

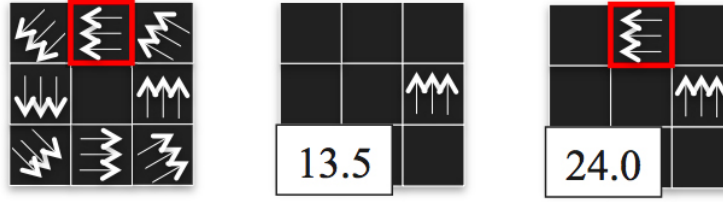


(a) Flow stimulus. (b) Singles stimulus.

Figure 5.8: Example of hotspot features. (a) Flow stimulus with unknown firing rate response to predict. (b) Singles stimulus shown in an O-configuration hotspot segment corresponding to the flow stimulus O-configuration hotspot segment. The hotspot contribution for this segment is simply its recorded firing rate of 19.0 spikes/second. Since the direction of motion within the hotspot segment is  $180^\circ$ , the hotspot horizontal contribution feature is equal to  $\cos(180^\circ) * 19.0 = -19.0$  spikes/second, and the hotspot vertical contribution feature is equal to  $\sin(180^\circ) * 19.0 = 0$  spikes/second.

on recorded singles data. The doubles pairing response feature is simply the recorded firing rate response for the doubles stimulus consisting of the direction of motion within the non-hotspot segment under consideration and the direction of motion within the corresponding hotspot segment. An example of a doubles pairing response feature, and the associated doubles pairing response minus segment contribution feature, is shown in Figure 5.9. Note that we did not use the last two non-hotspot doubles features to predict doubles stimuli firing rate responses because that would entail using the model to make predictions for the same data used for training, which would allow it to make perfect predictions.

You may be wondering why we use features that involve multiplying or adding two values together when such operations seem to be unnecessary when



(a) Flow stimulus. (b) Singles stimulus. (c) Doubles stimulus.

Figure 5.9: Example of doubles pairing response features. (a) Flow stimulus with unknown firing rate response to predict. (b) Singles stimulus shown in the non-hotspot segment for which we are determining the doubles pairing response. The non-hotspot contribution for this segment is simply its recorded firing rate of 13.5 spikes/second. (c) Doubles stimulus consisting of the pairing between the non-hotspot segment and hotspot segment. The doubles pairing feature value is the same as the firing rate for this doubles stimulus, which is 24.0 spikes/second. The value of doubles pairing response minus segment contribution feature is 10.5 spikes/second. Note that the directions of motion within the activate segments in the singles and doubles stimuli are the same as those shown within the corresponding flow stimulus segments.

employing LR. For example, if multiplying the hotspot contribution by the sine of the hotspot direction results in value that can be used to accurately model the response data then one would think that LR would assign appropriate weights to the hotspot contribution feature and a feature relating to the sine of the hotspot direction in order to form such a value. Preliminary tests showed that was not the case when such features were provided to LR, rather better results were produced by enforcing the multiplicative or additive relationship between certain features and sub-features.



## 5.12 Feature Sets

We combined subsets of the hotspot features and subsets of the non-hotspot features to form various feature sets. In turn we provided the features of each set as inputs to the LR process, one feature set per trial, to determine which feature set could be used to make the most accurate firing rate response prediction. Table 5.3, 5.4, and 5.5 list the various types of feature sets used. Notice that all of the feature sets for a given type share the same hotspot features. Only the non-hotspot features for a given type are different. Also notice that the non-hotspot features for feature sets 1 through 5 are the same for all of the types. By no means do the feature sets listed include all of the possible combinations of features listed in Table 5.1 and 5.2. The features in each set were hand-picked so that we could make certain conclusions based on the results. For example, no information about the direction of motion within the hotspot is included in type 1 feature sets. Since we assume that neuron behavior acts according to opponent organization, which is heavily dependent on direction of motion within hotspots, we anticipate that type 1 feature sets will produce the worst prediction results. Similarly, we assume that the direction of motion within non-hotspot segments can be used for prediction, and thus we anticipate that the first feature set of each type will produce poor prediction results.

Types 1, 2, and 3 form combinations of hotspot features that include magnitudinal hotspot contributions, directional hotspot contributions, and both, respectively. Similarly, feature sets 1, 4, and 5 do the same for non-

Type 1: Hotspot Features	
hotspot contrib.	
Type 1: Non-Hotspot Features	
Set 1	seg.contrib.
Set 2	seg. contrib. doubles pairing response
Set 3	seg. contrib. doubles pairing response – seg. contrib.
Set 4	sin seg. dir. * seg. contrib. cos seg. dir. * seg. contrib.
Set 5	seg. contrib. sin seg. dir. * seg. contrib. cos seg. dir. * seg. contrib.

Table 5.3: Type 1 feature sets.

Type 2: Hotspot Features	
sin hotspot dir. * hotspot contrib. cos hotspot dir. * hotspot contrib.	
Type 2: Non-Hotspot Features	
Set 1	seg.contrib.
Set 2	seg. contrib. doubles pairing response
Set 3	seg. contrib. doubles pairing response – seg. contrib.
Set 4	sin seg. dir. * seg. contrib. cos seg. dir. * seg. contrib.
Set 5	seg. contrib. sin seg. dir. * seg. contrib. cos seg. dir. * seg. contrib.

Table 5.4: Type 2 feature sets.

Type 3: Hotspot Features	
hotspot contrib. sin hotspot dir. * hotspot contrib. cos hotspot dir. * hotspot contrib.	
Type 3: Non-Hotspot Features	
Set 1	seg.contrib.
Set 2	seg. contrib. doubles pairing response
Set 3	seg. contrib. doubles pairing response – seg. contrib.
Set 4	sin seg. dir. * seg. contrib. cos seg. dir. * seg. contrib.
Set 5	seg. contrib. sin seg. dir. * seg. contrib. cos seg. dir. * seg. contrib.

Table 5.5: Type 3 feature sets.

hotspot features. In theory, if broad predictions can be made using magnitudinal features, then finely-tuned predictions may be made using directional features. Thus, including both magnitudinal and directional hotspot features in the same set may allow for better prediction results. Feature sets 2 and 3 include doubles pairing features. During preliminary tests the doubles pairing response features led to poor prediction results, thus we decided not to compose feature sets with both doubles pairing and non-hotspot directional features.

Since there is one X-configuration hotspot and one O-configuration hotspot the actual number of hotspot features per set is twice the number of hotspot features shown in Table 5.3, 5.4, and 5.5. Similarly, since there are 7 non-hotspots the actual number of non-hotspot features per set is 7 times the number of non-hotspot features shown in those tables. Therefore, the feature set with the most features is type 3 feature set 5 (T3-FS5), with 27 features. Since there are only 16 recorded flow data points we can use for training, we want to ensure that number of degrees of variation available to the LR technique is equal to or less than 16 to avoid over-fitting and ensure model generality. To accomplish this we discard certain low-impact non-hotspot features.

One approach would be to incrementally train on all features, discard the one assigned the lowest weight, and then train again on the remaining features, repeating the process until we are reach the desired number of features. Since all of the features are initially used for training, and assigned weights based on their complex interactions, this approach does not guarantee

that the most optimal features will not be discarded. Features are assigned different weights based on the other features in the set. Removing one feature may have a drastic effect on weight assigned to another feature. Thus, the initial features discarded may have been assigned the highest weights if they were retained and other features were discarded instead. Also, a low feature weight does not necessarily imply low feature significance because feature values were not normalized to comparable scales across feature types. Doing so would destroy the inherent relationship between different types of features.

In order to ensure optimal feature selection, we exhaustively compose all possible combinations of non-hotspot features per feature set and select the best one based on performance. Consider feature set 4, which has 14 non-hotspot features. An exhaustive combination of these features would result in  $2^{14}$  possible combinations. Note that some of these combinations omit only certain features from a non-hotspot while leaving other features for that non-hotspot intact. Other combinations omit all of the features from a non-hotspot, thus any firing rate predictions performed using such a combination will not consider activity within that non-hotspot, essentially considering it as noise and ignoring it. We call these segmental combinations. Since there are 7 non-hotspots, there are a total of  $2^7$  possible combinations of non-hotspot segments to choose from, and therefore there are  $2^7$  segmental combinations. Note that there is one combination which does not omit any non-hotspot features. To recap, LR results will be categorized into 3 types of feature combinations: exhaustive, segmental, and all-inclusive combinations.

### 5.13 System Design

The overall design of our system incorporates both the SGMM, trained using the CA, as well as the LR technique using non-linear hotspot and non-hotspot input features. The SGMM is used to model singles data in order to predict firing rate response values for cardinal and non-cardinal directions of motion. These values are then used to calculate hotspot and non-hotspot features. Doubles pairing features are calculated using recorded doubles data. These features are provided as inputs to the LR technique where they are assigned weights to form a linear model capable of predicting response values for doubles and flow stimuli. Although the same linear model can be trained to predict both doubles stimuli and flow stimuli, as shown in Figure 5.10, preliminary tests have shown that we achieve better results using a separate model for doubles stimuli and flow stimuli, as shown in Figure 5.11 and 5.12.

The SGMM itself is capable of predicting singles stimuli responses while the intent of the linear model is to capture interaction effects between hotspots and non-hotspots as well as between pairs of non-hotspots in doubles and flow stimuli. Since singles stimuli consist of only one active segment, there is no interaction effect, so a linear model is not used for singles stimuli response prediction. As discussed in Section 5.9, another approach to modeling doubles stimuli responses is to use separate feature spaces, one for each hotspot segment and hotspot direction combination. We can then train a separate SGMM unique to each feature space. The combination of separate SGMMs forms a hierarchical model, as shown in Figure 5.13.

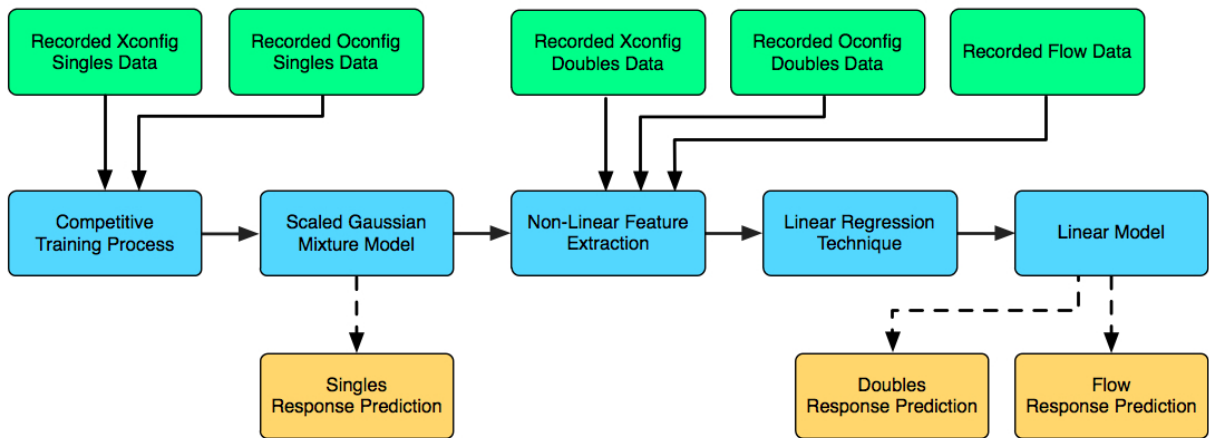


Figure 5.10: Possible LR system for doubles and flow prediction.

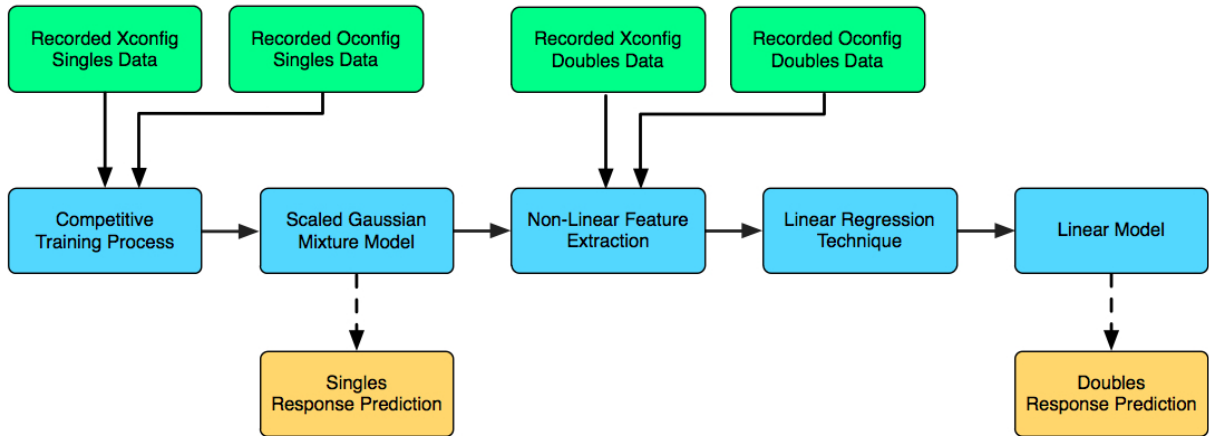


Figure 5.11: LR system for doubles prediction.

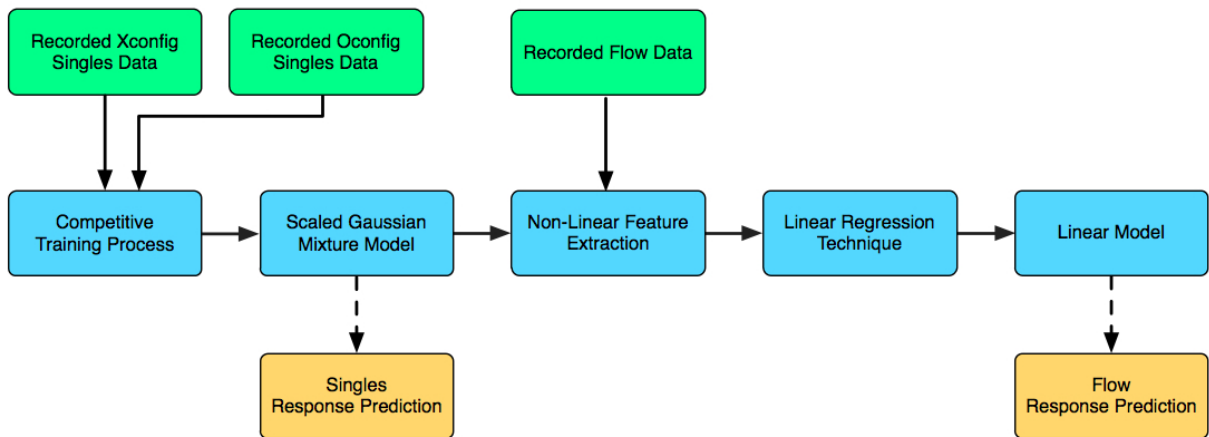


Figure 5.12: LR system for flow prediction.



Figure 5.13: CA system for hierarchical doubles prediction.



## Chapter 6

### Results

#### 6.1 Probability Function and Directional Covariation Test Case

A series of tests were performed on neuron 712R02 in order to determine the impact of using the PDF versus the CDF for the SGMM Gaussian probability function and the impact of training models with and without directional covariation. Four types of models were trained using the GA: PDF with directional covariation, PDF without directional covariation, CDF with directional covariation, and CDF without directional covariation. Each stage of GA training began by randomly generating 500 models and selecting the top 50 models to form the initial population to undergo genetic training, which lasted for 30 generations. Staged training ceased when either 10 Gaussians were added to the model, the residual error on the training set dropped below 1.0 spikes/second, or attempting to add another Gaussian to the model caused an increase in training set residual error. Rather than always allowing the model to train to convergence, a fixed number of Gaussians was chosen as a cutoff point to stop training. The later a Gaussian is added to the model the less value it serves because it may result in a trivial performance improvement in terms of residual error reduction. For example, a 10-Gaussian model with



Figure 6.1: Example singles data GA test set. Note that each of the 4 data points correspond to a different direction of cardinal motion and different segment.

good performance could take on another 10 Gaussians before convergence, but that often results in over-fitting the training data and poor generalization.

A form of random sub-sampling (i.e. holdout) validation was performed by partitioning the singles data set for the neuron into a training set and test set. Specifically, 4 out of the 36 data points were selected for testing and the other 32 were used for training. The test data points were randomly selected with the constraint that no two of the data points in the test set could share the same direction or segment. An example test set is shown in Figure 6.1. This constraint was necessary to prevent large gaps from forming in the (x, y, dir.) space due to the lack of training data in neighboring areas, which would in turn make it difficult to fit Gaussian distributions to that empty space. The constraint forces a more uniform distribution of test data across the (x, y, dir.) space. The best performing model was determined based on each model's test set MSE. A total of 10 trials were conducted for each of the 4 model types. The model generated by each trial was validated using a different training set and test set partition. The same training set and test set partitions were used to validate the 10 models for each model type so that they could be compared. Only one model was trained per partition per model type.

Note that we do not repeat validation multiple times using different partitions of training data and test data on the same model, such as if we had decided to employ K-fold cross-validation. This is because each repeat run of the training algorithm for each data partition could potentially generate a completely different model due to the randomized nature of the GA. Each model may have very different properties and perform significantly better or worse than other models trained using different partitions of the same data set. In other words, the GA is unstable in the sense that there is a high probability that it will not generate a similar model when trained using similar data. Thus, a repetitive cross-validation technique would serve very little value in understanding the performance of any particular model trained by the GA. A consequence of our sub-sampling validation technique is that only certain data points are selected for testing purposes and the firing rates associated with those data points may be easier to predict than those used for training. Thus, our validation technique provides an optimistic view of how the SGMM generalizes to novel data.

Results are shown in Table 6.1 through 6.4 as well as in Figure 6.2 and 6.3. The standard error range of values for each data point for neuron 712R02 is shown by the error bars in Figure D.1 in Appendix D. Note that the mean squared error (MSE) and mean absolute error (MAE) values shown throughout this chapter retain their statistical meanings, which are not to be confused with the various fitness functions discussed so far. Mean residual error (MRE) is the average residual error calculated over all of the data points in a set.

712R02 Singles Results, PDF with Directional Covariation (MaxFiringRate: 40.25)							
	Train			Test			
Partition	MSE	MAE	MRE	MSE	MAE	MRE	NumGauss
1	8.1383	1.4579	0.6402	5.6144	2.0051	0.7041	10
2	4.9938	1.3747	0.5491	120.7957	9.1424	7.8820	10
3	16.9058	1.7270	1.0473	12.7956	2.8091	1.7537	10
4	31.3561	1.6733	0.9616	12.0696	2.8453	2.0341	10
5	1.5471	0.9190	0.0972	887.3463	18.7371	17.4958	10
6	1.9756	0.9555	0.1577	9.3389	2.7927	1.7375	10
7	2.7222	1.0781	0.2350	15.4920	3.3512	2.1776	10
8	41.648	1.9242	1.2735	16.6279	3.7213	1.9970	10
9	3.4481	1.3927	0.4067	304.1403	13.1983	11.4054	10
10	4.1727	1.1871	0.5186	2639.4916	30.3022	28.4353	10

Table 6.1: Neuron 712R02 GA singles training results for PDF models with directional covariation.

712R02 Singles Results, PDF with No Directional Covariation (MaxFiringRate: 40.25)							
	Train			Test			
Partition	MSE	MAE	MRE	MSE	MAE	MRE	NumGauss
1	9.0737	1.9376	0.9340	25.0070	4.3572	2.7270	10
2	18.8450	2.9985	1.7787	12.2050	2.7129	1.7460	3
3	13.1284	2.2843	1.2753	248.4888	11.2771	9.8325	7
4	18.5544	2.0896	1.0277	126.7358	9.0617	8.0867	10
5	6.7042	1.7817	0.7689	414.6871	13.5773	12.2208	10
6	5.7096	1.8044	0.5840	282.5714	10.6609	9.6056	10
7	5.7742	1.5041	0.6297	1452.8101	22.9510	21.7618	10
8	28.2779	2.0810	1.1449	212.5527	10.9714	9.4163	10
9	10.4522	2.2593	1.1815	1.8012	1.1380	0.1939	10
10	6.6280	1.9408	0.9025	258.6976	11.8270	10.1878	10

Table 6.2: Neuron 712R02 GA singles training results for PDF models with no directional covariation.

712R02 Singles Results, CDF with Directional Covariation (MaxFiringRate: 40.25)							
	Train			Test			
Partition	MSE	MAE	MRE	MSE	MAE	MRE	NumGauss
1	3.9233	1.1662	0.3208	11.8513	3.0581	1.6188	10
2	2.6602	0.8345	0.2233	46.4418	5.1399	3.8998	10
3	4.8896	1.4944	0.5200	147.3598	8.2655	7.3922	10
4	7.5866	0.9704	0.4580	83.3804	8.1081	7.1332	10
5	2.6429	0.8847	0.2648	30.8776	5.1521	3.5844	10
6	1.7817	0.9531	0.1181	20.1529	4.0392	2.9840	10
7	2.6711	1.1328	0.2476	19.9718	3.1204	1.9957	10
8	10.5071	1.6659	0.6986	11.7622	2.6281	1.5392	8
9	1.5865	0.9115	0.1569	51.1434	6.7253	4.8584	10
10	2.7048	0.9823	0.3044	56.2407	5.7216	3.9480	10

Table 6.3: Neuron 712R02 GA singles training results for CDF models with directional covariation.

712R02 Singles Results, CDF with No Directional Covariation (MaxFiringRate: 40.25)							
	Train			Test			
Partition	MSE	MAE	MRE	MSE	MAE	MRE	NumGauss
1	6.1922	1.9380	0.6729	178.5464	10.2873	8.7236	10
2	18.0101	2.9398	1.6263	966.4842	17.8864	16.6661	4
3	13.1149	2.1961	1.2304	581.2520	15.7205	14.3743	7
4	5.7962	1.6325	0.6256	135.3712	10.6365	9.6615	10
5	6.2153	1.7498	0.7303	99.0757	8.2359	6.6682	10
6	8.3673	2.0977	0.9566	38.2273	5.7385	4.6833	10
7	7.5690	1.9356	0.7716	31.2921	4.2813	3.1910	9
8	37.3879	2.8805	1.6452	35.0645	4.8830	3.2929	6
9	20.5716	3.3542	2.0349	8.9960	2.1914	0.7690	4
10	6.4302	1.7258	0.7203	239.3099	13.8146	11.9478	10

Table 6.4: Neuron 712R02 GA singles training results for CDF models with no directional covariation.

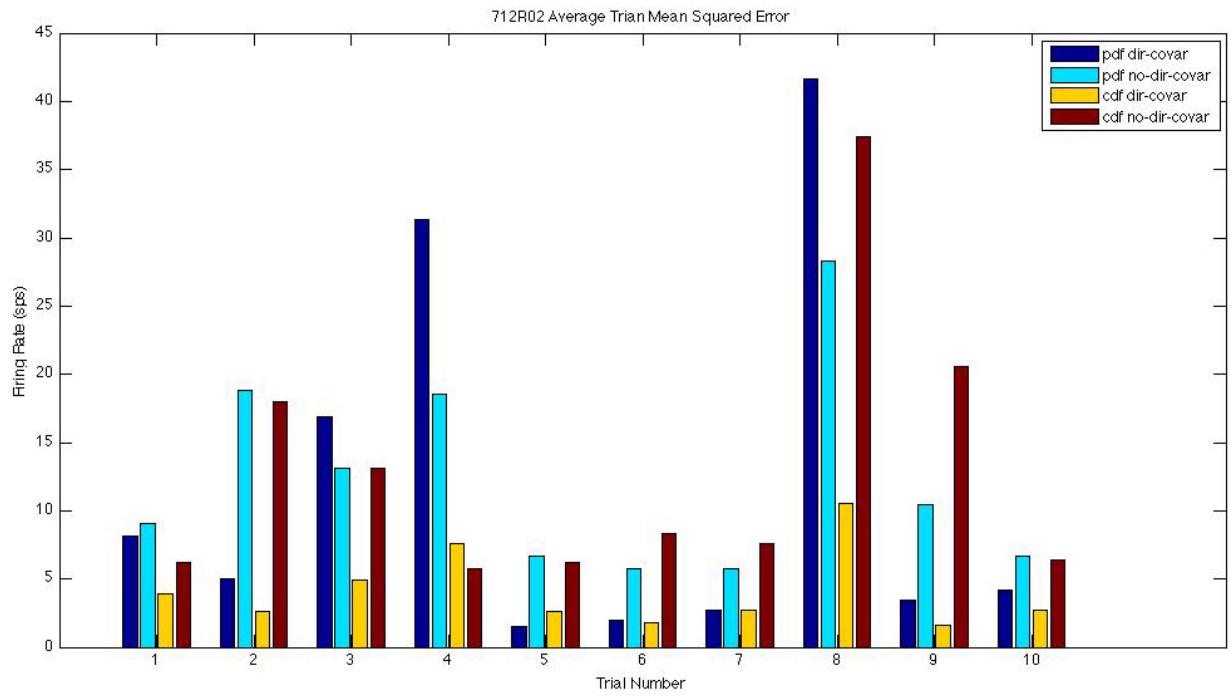


Figure 6.2: Neuron 712R02 GA singles prediction MSE training performance.

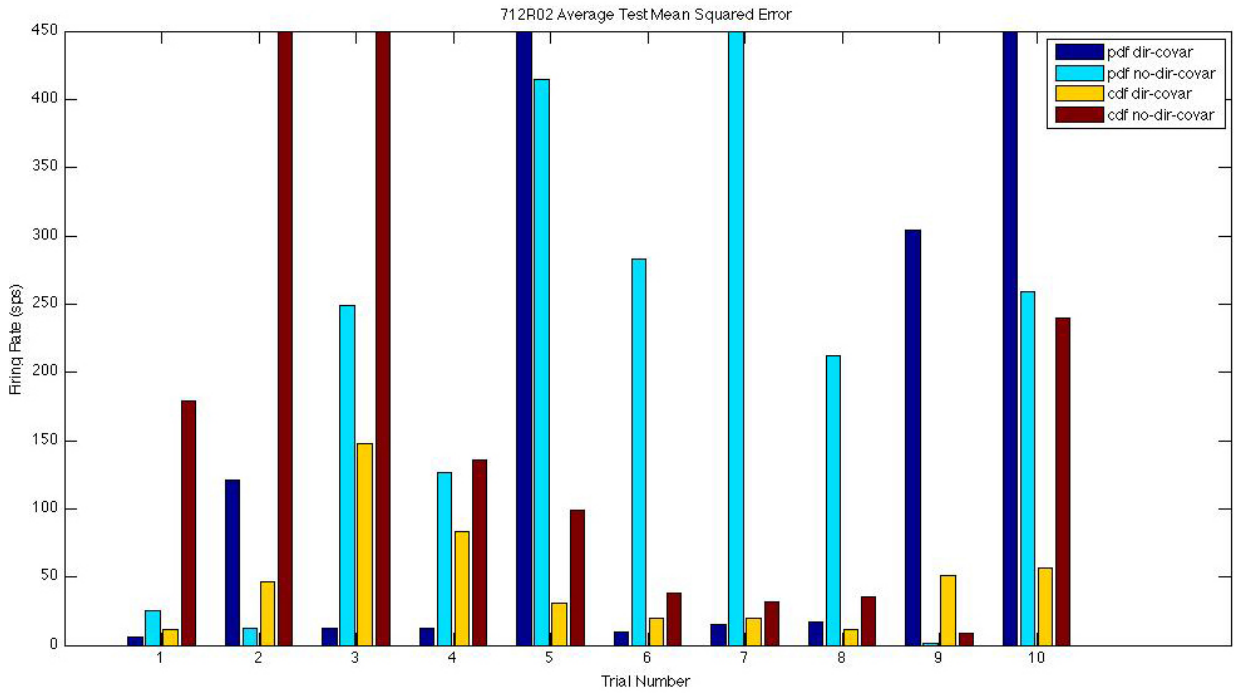


Figure 6.3: Neuron 712R02 GA singles prediction MSE test performance.

According to the results shown in Figure 6.2, 7 out of 10 PDF models and 9 out of 10 CDF models had better MSE training performance results when directional covariation was allowed versus when it was not allowed. According to the results shown in Figure 6.3, 6 out of 10 PDF models and 9 out of 10 CDF models had better MSE test performance results when directional covariation was allowed versus when it was not allowed. Additionally, when directional covariation was allowed, 9 out of 10 CDF models had better MSE training performance than the corresponding PDF models and 5 out of 10 CDF models had better MSE test performance than the corresponding PDF models. Based on this information it appears that CDF models are capable of being trained to fit the data more precisely than PDF models; however, it is inconclusive whether or not CDF models are capable of generalizing more accurately than PDF models. Although CDF models are an appealing consideration, they simply take too long to generate and validate to make them a reasonable choice without the availability of a super computer or grid computing system. As mentioned in Section 5.4, CDF models take an average of 9-10 times longer to generate and validate than PDF models. For this reason we employ PDF models in the experiments discussed in Section 6.2 and 6.3.

## **6.2 Competitive Algorithm Applied to Singles Data**

The CA algorithm was applied to the singles data to train a SGMM for 5 neurons. Due to the randomized nature of the GA component of the CA, 50 trials were conducted for each neuron and the top performing SGMM among

those trials was selected. Repeat trials were necessary to train a reasonable model because not all of the models trained performed well, which is caused by the unstable nature of the GA. Each stage of GA training began by randomly generating 500 models and selecting the top 50 models to form the initial population to undergo genetic training, which lasted for 30 generations. Staged training ceased when either 20 Gaussians were added to the model, the residual error on the training set dropped below 1.0 spikes/second, or attempting to add another Gaussian to the model caused an increase in training set residual error. The same random sub-sampling validation procedure described in Section 6.1 was used to determine how well each model generalizes. Again, a fixed number of Gaussians was chosen as a cutoff point to stop training in order to prevent over-fitting the training data.

The validation results for singles data are shown in Table 6.5 and Figure 6.4 as well as in Figure D.1 through D.4 in Appendix D. The singles stimuli numbering scheme is shown in Appendix B. The Gaussians trained for the SGMM use the PDF as the probability function and allow for directional covariation. Note that the table and figures show results for the best performing SGMM for the neuron in question. The standard error range of values for each data point is shown by the vertical error bars. Note that the standard error range of values for many of the data points for neuron 819R10 is quite large. This implies that the recorded data is noisy or inaccurate, which makes it difficult to accurately train a model to fit that data and may explain why a large number of Gaussians were trained for that neuron using the GRA instead



Singles Results, PDF with Directional Covariation						
	Train			Test		
	MSE	MAE	MRE	MSE	MAE	MRE
712R02	1.3060	0.6214	0.1392	4.9922	1.9097	1.0127
819R09	0.3305	0.36001	0.0282	5.1452	2.1902	0.9151
819R10	66.4277	5.3915	1.1068	236.2340	12.7746	11.2329
819R32	0.0722	0.1709	0.0106	2.5213	1.4490	0.6276
819R64	0.1920	0.3087	0.0291	0.7069	0.7215	0.4065
	MaxFiringRate	NumGauss	NumGenetic	NumGreedy		
712R02	40.25	12	6	6		
819R09	17.40	12	11	1		
819R10	128.33	13	2	11		
819R32	21.00	20	16	4		
819R64	16.50	7	6	1		

Table 6.5: CA singles prediction results.

of the GA. Figure 6.5 shows an example of how the CA training procedure attempts to reduce the various error and fitness values with the addition of each new Gaussian to neuron 819R32’s SGMM. Figure 6.6 and 6.7 show the placement of Gaussians in the (x, y, dir) feature space of that model. Overall, the CA is able to effectively model the singles data for each of the neurons.

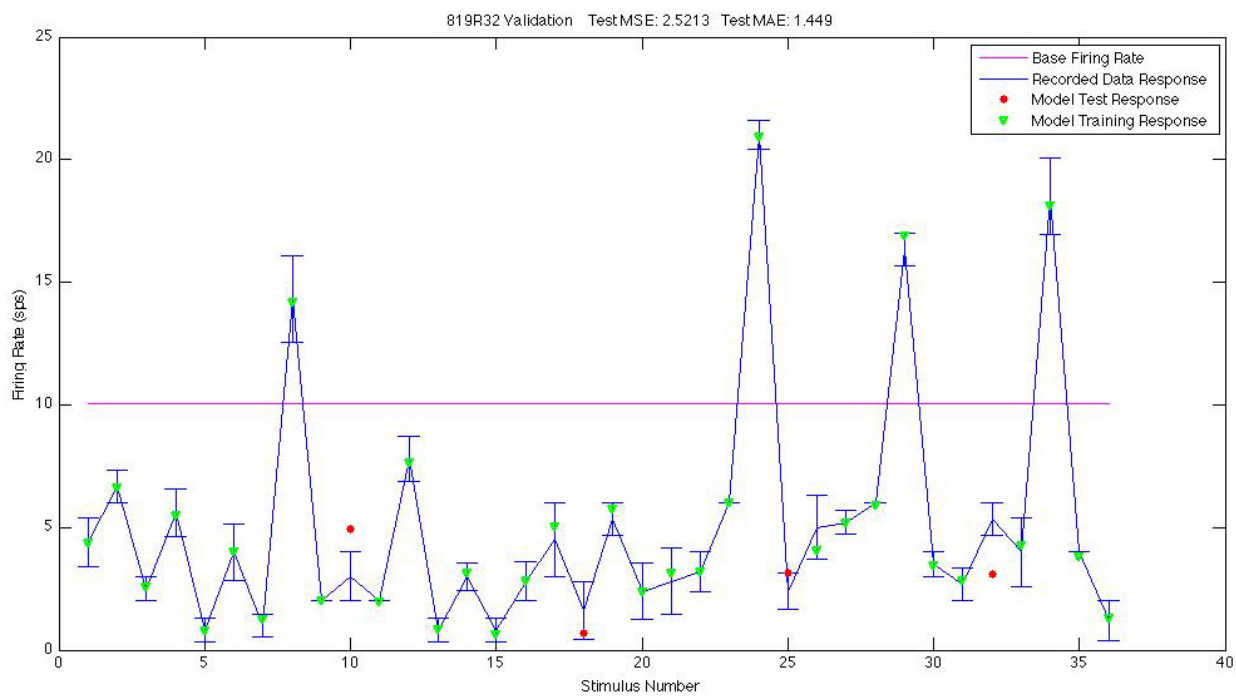


Figure 6.4: Neuron 819R32 CA singles prediction.

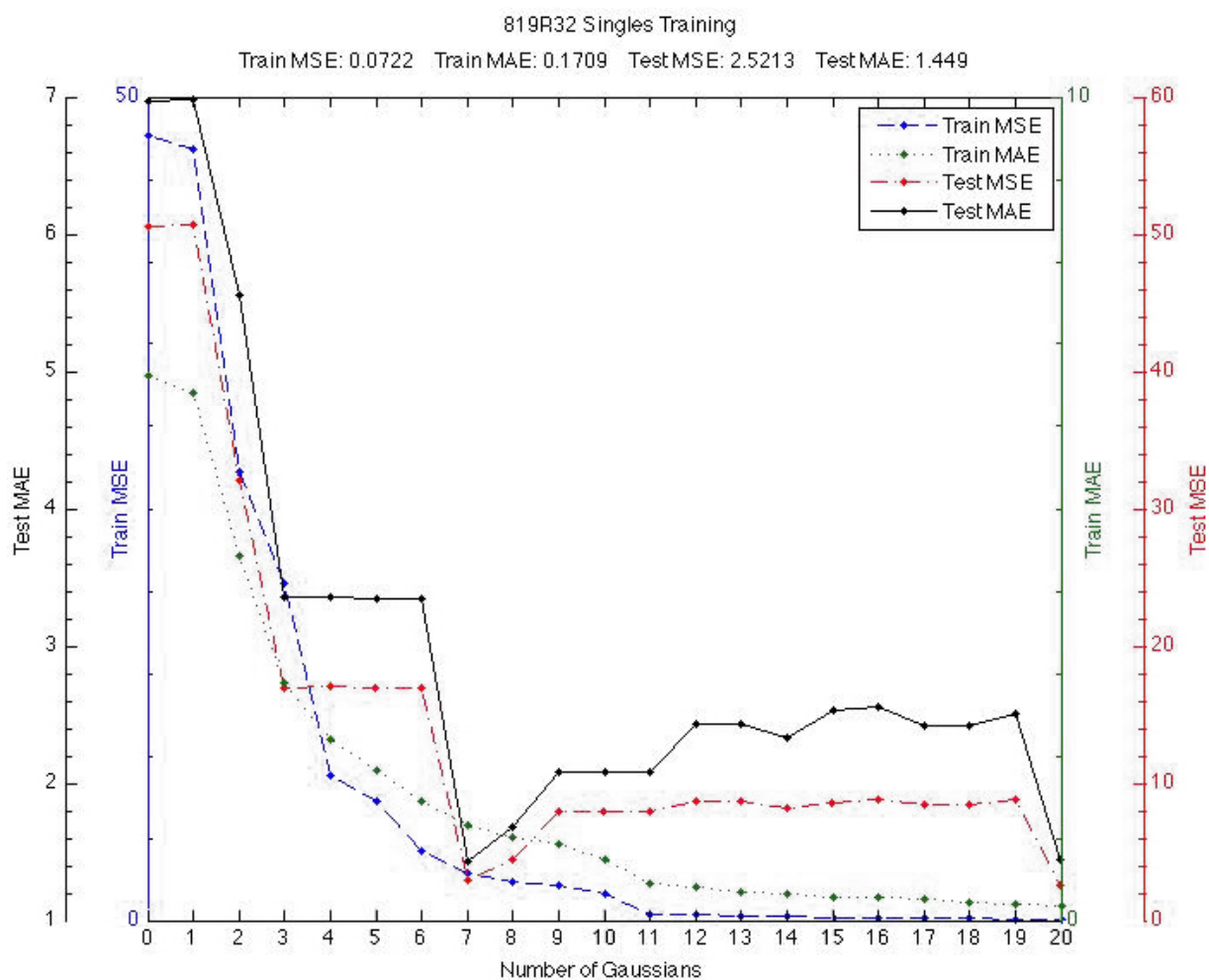


Figure 6.5: Neuron 819R32 CA singles training history. Training error decreases very quickly with the addition of the first 4 Gaussians and then begins to level off as more Gaussians are added to the model. Test error begins to increase with the addition of the 8th Gaussian but then drops to a reasonable value with the addition of the 20th Gaussian.

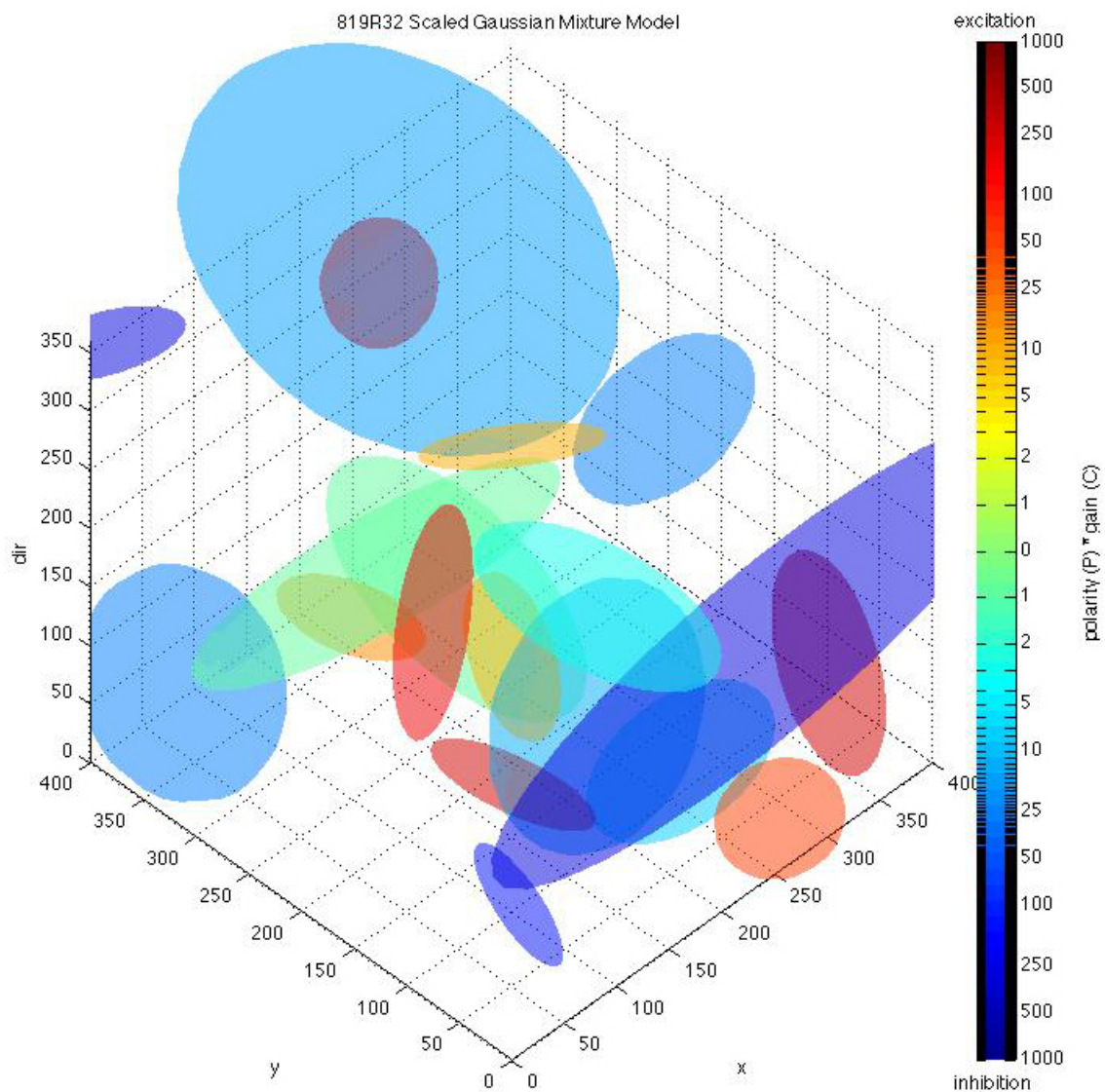
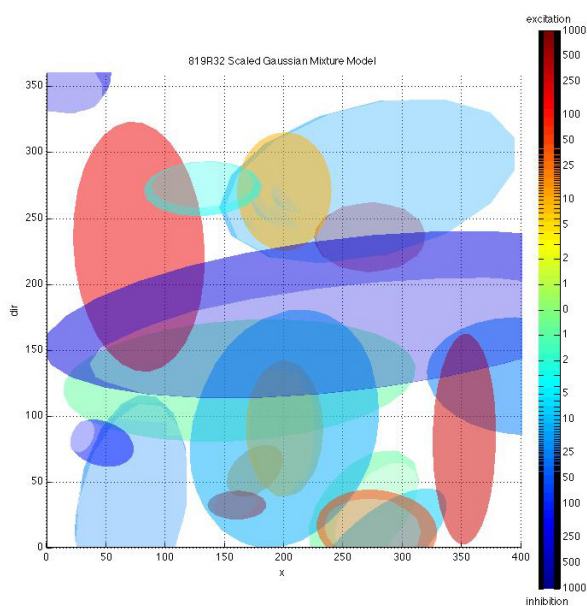
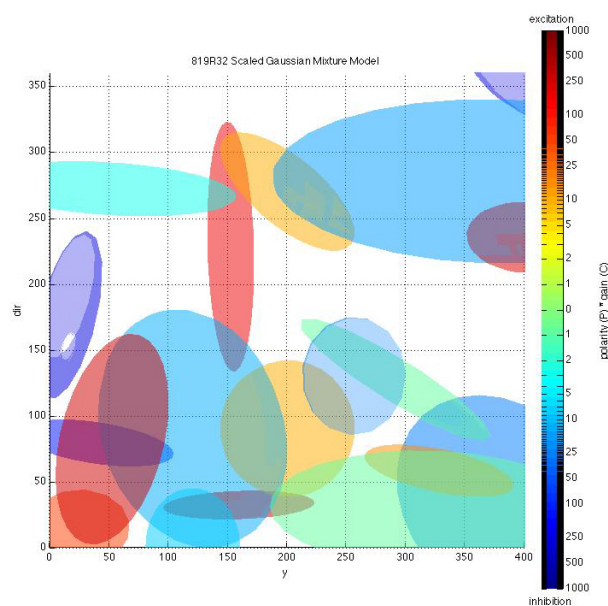


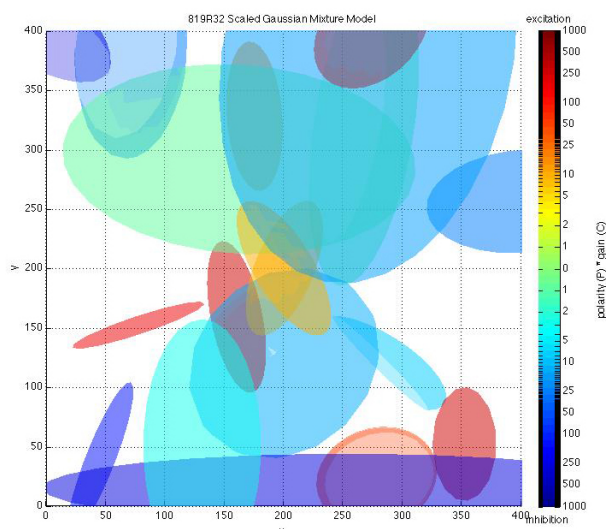
Figure 6.6: Neuron 819R32 SGMM Gaussians in (x, y, dir) feature space. 3-D view. Gaussians shown up to the first standard deviation. X-axis and Y-axis units correspond to those shown in Figure 5.1(a). Dir-axis units are in degrees. Color represents excitatory or inhibitory behavior according to the value of  $P * C$  in Equation 5.6. Note that there is considerable overlap between Gaussians at standard deviations 2 and higher (not shown). The Gaussian rollover between  $0^\circ$  and  $360^\circ$  is not shown, but is accounted for in the algorithm.



(a) (x, dir) view.



(b) (y, dir) view.



(c) (x, y) view.

Figure 6.7: Neuron 819R32 SGMM Gaussians in (x, y, dir) feature space. 2-D views. Gaussians shown up to the first standard deviation.

### 6.3 Competitive Algorithm Applied to Doubles Data

As discussed in Section 5.9, and shown in Figure 5.13, the CA was applied to doubles data by subdividing the data into 8 separate RF feature spaces consisting of the 4 cardinal directions of motion for each of the 2 hotspot segments. Thus, all of the data points within each feature space correspond to doubles stimuli that share the same hotspot segment and direction of motion within that segment. This subdivision is justified because we strongly believe that motion within the hotspot segment has a major impact on the firing rates produced for non-hotspot segment motion pairings with that hotspot segment motion. This is known as the interaction effect and is related to opponent organization, as discussed in Section 3.4.

One SGMM was trained for each of the 8 feature spaces using the same approach discussed in Section 6.2 to train SGMMs for singles data. The combination of SGMMs for each neuron forms a hierarchical model. A hierarchical model was trained for each of the 5 neurons. Although each doubles stimulus consists of hotspot and non-hotspot segment motion, the hotspot segment motion is encoded in the feature space, and therefore does not need to be taken into further consideration when training the SGMM using the CA. We are left with non-hotspot segment motion, which takes the same form as singles stimuli. Therefore, training on each doubles hotspot subspace data set is the same as training on singles data. Since there are 4 X-configuration non-hotspot segments and 4 cardinal directions of motion there are a total of  $4 * 4 = 16$  data points for each X-configuration hotspot subspace. Similarly, since there are

3 O-configuration non-hotspot segments there are a total of  $3 * 4 = 12$  data points for each O-configuration hotspot subspace.

The CA algorithm was applied to the doubles data to train 8 separate SGMMs for each neuron. Each doubles SGMM was trained in the same way that each singles SGMM was trained using the CA, as described in Section 6.2. Additionally, the same random sub-sampling validation procedure described in Section 6.1 was used to determine how well each model generalizes. The only difference is that 2 test points were selected for each doubles subspace model compared to the 4 test points selected for each singles model. The number of test points was reduced to scale with the smaller number of data points per doubles subspace model (16 or 12) compared with the number of data points per singles model (36). The validation results for each of the separate subspaces for each neuron are shown in Table 6.6 as well as in Table E.1 through E.4 in Appendix E. The combined results for each neuron shown in Table 6.7 were calculated by the hierarchical model over the entire doubles data set. The hierarchical model simply employs the appropriate subspace model based on the direction of motion through the neuron’s hotspot segment for each doubles stimulus. The data points used to test the hierarchical model are simply the collection of data points used to test each subspace model for that neuron.

Each of the 8 separate subspace results, as well as the combined results, are shown for neuron 819R64 in Figure 6.8. The combined results for the other neurons are shown in Figure E.1 through E.4 in Appendix E. The

doubles stimuli numbering scheme for each neuron is shown in Appendix C. Similar to the CA singles data experiments, the Gaussians trained for the CA doubles data experiments use the PDF as the probability function and allow for directional covariation. The values calculated in the tables in figures retain the same meaning they did for the CA singles data experiments and only results for the best performing SGMM for the neuron and subspace in question are shown. Once again it proves difficult to accurately train a model to fit data points for neuron 819R10. Overall, the CA is able to effectively model the doubles data for each of the neurons by subdividing the RF feature space.



Neuron 819R64 Doubles Subspace Results, PDF with Directional Covariation						
	Train			Test		
	MSE	MAE	MRE	MSE	MAE	MRE
Xconfig. 0°	1.3655	0.7687	0.3871	0.7574	0.6882	0.0778
Xconfig. 90°	0.2571	0.3299	0.0643	0.7939	0.7202	0.1325
Xconfig. 180°	0.3546	0.4894	0.0660	0.4461	0.6280	0.4277
Xconfig. 270°	0.5043	0.4567	0.0103	0.2440	0.4816	0.0000
Oconfig. 0°	0.4097	0.4720	0.0595	0.1939	0.3498	0.0000
Oconfig. 90°	0.1207	0.2642	0.0937	0.2533	0.4557	0.0000
Oconfig. 180°	0.0850	0.2171	0.0000	0.5080	0.7125	0.0313
Oconfig. 270°	0.8507	0.5669	0.0000	0.3991	0.5709	0.0000
	MaxFiringRate	NumGauss	NumGenetic		NumGreedy	
Xconfig. 0°	22.00	4	2		2	
Xconfig. 90°	16.50	4	2		2	
Xconfig. 180°	10.75	2	1		1	
Xconfig. 270°	18.50	4	1		3	
Oconfig. 0°	22.00	4	1		3	
Oconfig. 90°	10.00	3	2		1	
Oconfig. 180°	3.50	3	1		2	
Oconfig. 270°	18.00	3	1		2	

Table 6.6: Neuron 819R64 CA doubles subspace prediction results.

Doubles Results, PDF with Directional Covariation						
	Train			Test		
	MSE	MAE	MRE	MSE	MAE	MRE
712R02	1.1549	0.6123	0.0591	0.6791	0.2260	3.6577
819R09	0.8867	0.5403	0.1308	0.4750	0.2013	6.1481
819R10	147.1413	6.3360	1.0254	16.3040	1.2544	46.6225
819R32	0.5878	0.4317	0.1059	0.1698	0.1097	4.1349
819R64	0.4411	0.3913	0.0929	0.0642	0.0823	1.3385
	MaxFiringRate	NumGauss	NumGenetic		NumGreedy	
712R02	41.25	29	15		14	
819R09	44.67	30	16		14	
819R10	164.00	31	10		21	
819R32	33.80	31	15		16	
819R64	22.00	27	11		16	

Table 6.7: CA doubles prediction results. Combination of subspace results.

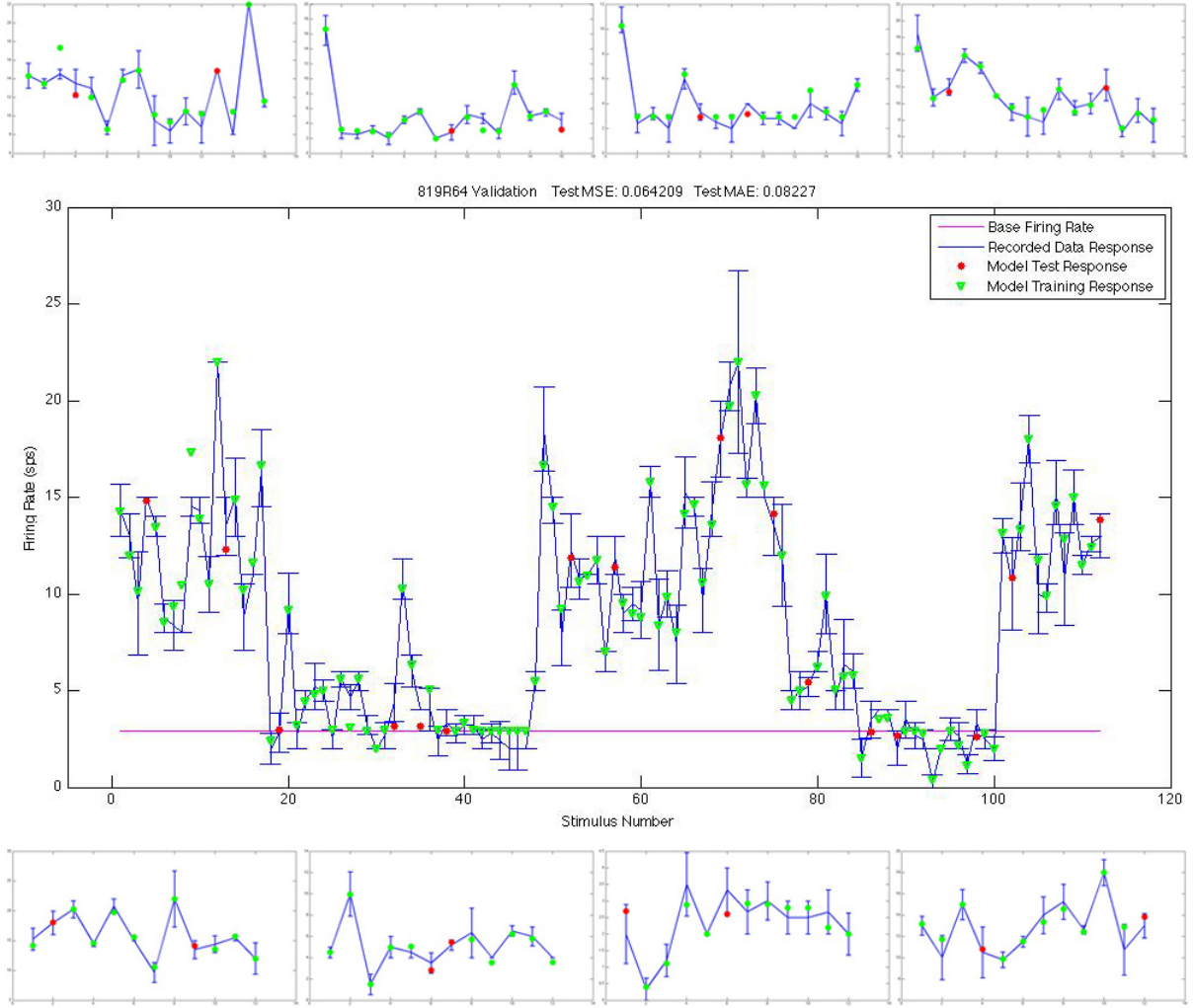


Figure 6.8: Neuron 819R64 CA doubles prediction. Smaller plots show separate subspaces corresponding to hotspot segment and direction combinations. Center plot shows the result of combining all of the separate subspace results together. Top row (left to right): Xconfig. 0°, Xconfig. 90°, Xconfig. 180°, Xconfig. 270°. Bottom row (left to right): Oconfig. 0°, Oconfig. 90°, Oconfig. 180°, Oconfig. 270°.

## 6.4 Linear Regression Technique Applied to Flow Data

Combinations of the feature sets shown in Table 5.3, 5.4, and 5.5 were provided as inputs to the LR technique for 5 neurons. Training was performed on all of the segmental combinations for each feature set and on all of the exhaustive combinations for those feature sets with less than 3 non-hotspot features. Training on each of the  $2^{21}$  combinations for feature set 5 would be intractable and therefore wasn't performed. Leave-one-out cross-validation (LOO-CV) was performed to determine how well the models produced by the LR technique generalize. Specifically, LOO-CV was performed on the flow data by partitioning the 16 recorded data points into 16 partitions of equal size and training and testing on each. Each partition was composed of 15 training data points and 1 test data point. The test point used in each partition was unique. Error values were averaged over all of the partitions.

Validating the linear model in this way provides a pessimistic view of how it generalizes because each recorded data point is validated in isolation, including those which may be the result of noise in the data collection process and do not follow the same trend as the rest of the data. Also note that LR was not calculated with a constant value. Preliminary tests showed that the use of a constant value produced models that did not perform as well as those without a constant value. The LR LOO-CV results for flow data are shown in Table 6.8, 6.9, and 6.10. Note that each table only shows the error values for the best performing combination for each feature set.

Flow Results, Combination of All Features in Set										
	712R02 Test		819R09 Test		819R10 Test		819R32 Test		819R64 Test	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
T1-FS1	3377.49	47.72	826.64	23.42	3699.55	47.04	116.49	8.89	72.09	7.31
T1-FS2	216525.74	326.72	1477.45	23.51	3710.27	51.20	182.18	12.32	1.31	0.97
T1-FS3	189537.79	312.57	1598.82	24.19	3855.54	52.30	112.23	8.70	481.14	18.38
T1-FS4	199893.38	377.50	8344.94	72.29	45030.43	155.92	4980.24	50.23	107.21	5.80
T1-FS5	20687.04	103.20	329.88	10.17	2070.57	34.03	671.74	20.71	127.97	10.32
T2-FS1	4998.06	54.42	926.34	24.05	521.23	18.40	266.13	14.04	71.47	6.90
T2-FS2	28088.21	124.27	5368.77	55.57	2989.71	45.25	79.53	7.04	147.41	7.34
T2-FS3	27117.52	124.70	4168.93	50.68	3747.23	51.12	38.33	5.45	132.83	10.01
T2-FS4	35211.92	157.79	2007.42	33.59	4452.92	46.65	158.73	8.95	39.60	4.53
T2-FS5	10464.85	73.67	857.57	19.38	815.22	23.36	214.02	12.35	19.12	3.30
T3-FS1	5395.15	57.68	4874.43	52.24	4918.39	51.08	72.37	6.90	770.55	21.37
T3-FS2	84671.50	220.10	2699.82	39.54	1933.44	39.56	126.50	9.75	220.20	11.98
T3-FS3	113593.05	234.66	2909.39	45.49	3939.81	55.25	125.85	9.50	179.26	10.19
T3-FS4	7505.15	67.83	2841.51	46.23	7736.56	77.91	314.81	15.46	64.49	6.87
T3-FS5	5189.09	57.89	662.44	20.75	1278.11	30.53	277.26	12.67	125.06	9.35
MaxFiringRate	101.67		88.80		91.50		39.20		32.25	

Table 6.8: LR flow prediction results using combination of all set features.

Flow Results, Best Segmental Combination of Features in Set										
	712R02 Test		819R09 Test		819R10 Test		819R32 Test		819R64 Test	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
T1-FS1	880.28	25.83	367.39	13.60	809.23	22.33	41.48	4.83	30.71	4.48
T1-FS2	1011.05	28.34	34.80	5.24	505.67	17.52	25.81	3.96	1.31	0.97
T1-FS3	1011.05	28.34	34.80	5.24	505.67	17.52	14.94	2.97	16.18	3.28
T1-FS4	733.66	22.87	86.87	5.82	794.77	20.93	62.23	5.35	15.37	3.36
T1-FS5	806.68	20.29	253.06	12.66	465.36	18.44	27.43	4.43	10.50	2.71
T2-FS1	1228.88	28.22	317.36	12.73	224.87	11.89	90.14	6.91	18.28	3.49
T2-FS2	296.45	14.57	200.47	10.87	316.85	13.91	19.97	3.94	27.94	4.65
T2-FS3	296.45	14.57	200.47	10.87	316.85	13.91	15.04	3.10	26.90	4.27
T2-FS4	1396.79	33.63	183.95	10.78	311.95	14.21	21.49	3.53	9.17	2.60
T2-FS5	58.11	4.60	56.61	6.13	131.96	8.49	40.03	5.19	0.88	0.64
T3-FS1	716.83	22.06	352.22	15.75	245.67	13.18	27.96	4.80	41.88	4.98
T3-FS2	268.16	13.93	91.36	7.92	406.38	16.34	5.08	1.70	6.87	2.18
T3-FS3	268.16	13.93	45.78	5.33	406.38	16.34	24.70	4.31	14.48	3.31
T3-FS4	25.54	4.66	169.21	11.21	486.64	18.47	0.63	0.61	11.18	2.32
T3-FS5	456.21	13.95	73.15	6.86	360.90	12.55	2.22	1.13	0.63	0.44
MaxFiringRate	101.67		88.80		91.50		39.20		32.25	

Table 6.9: LR flow prediction results using best segmental combination of set features.

Flow Results, Best Exhaustive Combination of Features in Set										
	712R02 Test		819R09 Test		819R10 Test		819R32 Test		819R64 Test	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
T1-FS1	880.2793	25.8301	367.3865	13.5999	809.2255	22.3327	41.4789	4.8297	30.7091	4.4821
T1-FS2	35.9046	4.8426	24.2123	4.0677	12.5351	2.4979	16.0815	3.3088	0.1561	0.3194
T1-FS3	37.1539	4.9069	0.0001	0.0066	163.3437	6.9477	1.4871	0.9671	5.8017	2.0685
T1-FS4	494.0317	17.6080	11.9362	2.7105	338.0913	14.4231	55.5731	5.5838	5.5544	1.6899
T1-FS5	-	-	-	-	-	-	-	-	-	-
T2-FS1	1228.8783	28.2157	317.3638	12.7321	224.8669	11.8877	90.1395	6.9139	18.2846	3.4948
T2-FS2	0.2368	0.2489	8.8847	2.5188	1.2334	0.5530	0.1157	0.2680	0.0267	0.1264
T2-FS3	0.2368	0.2489	0.0380	0.1254	1.2334	0.5530	0.2644	0.4267	0.0108	0.0698
T2-FS4	1.0058	0.6068	2.3934	1.0989	1.1891	0.8848	0.2271	0.3253	0.0111	0.0862
T2-FS5	-	-	-	-	-	-	-	-	-	-
T3-FS1	716.8264	22.0604	352.2177	15.7471	245.6691	13.1815	27.9608	4.8048	41.8847	4.9752
T3-FS2	0.3006	0.3241	0.0023	0.0220	0.1017	0.2202	0.0154	0.0727	0.0066	0.0645
T3-FS3	0.1389	0.3128	0.0156	0.1079	5.5356	1.9554	0.0204	0.1011	0.1381	0.2069
T3-FS4	0.0393	0.0700	0.0004	0.0129	0.0027	0.0345	0.0093	0.0393	0.1007	0.2225
T3-FS5	-	-	-	-	-	-	-	-	-	-
MaxFiringRate	101.67		88.80		91.50		39.20		32.25	

Table 6.10: LR flow prediction results using best exhaustive combination of set features.

Based on these results it is clear that specific exhaustive combinations of features in sets T2-FS2, T2-FS3, T2-FS4, T3-FS2, T3-FS3, and T3-FS4 produced models that are the most capable of predicting firing rate responses for flow data. Type 2 and type 3 feature sets include directional hotspot features, which supports our hypothesis that the direction of motion within hotspot segments plays a large role in predicting response values. Specifically, combinations of features in set T3-FS4 perform the best for 4 out of the 5 neurons in terms of MSE. Since T3-FS4 includes directional features for hotspots and non-hotspots, this result also supports our hypothesis that there is an interaction between hotspot and non-hotspot features according to the direction of motion within hotspot and non-hotspot segments. Plots of the best performing T3-FS4 combination for each neuron are shown in Figure 6.9 as well as in Figure F.1 through F.4 in Appendix F. The flow stimuli numbering

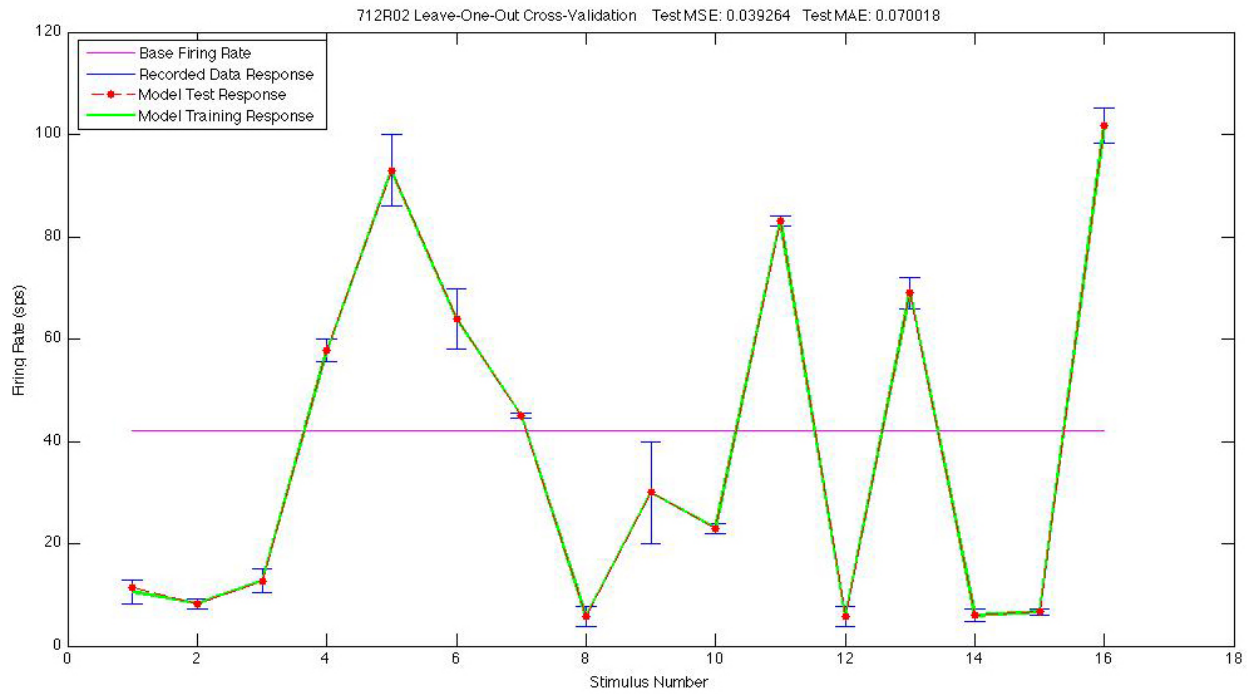


Figure 6.9: Neuron 712R02 LR flow prediction using best T3-FS4 combination.

scheme is shown in Figure 3.5. Note that the features for each neuron are not necessarily the same. In other words, the best performing T3-FS4 model for each neuron may consist of different non-hotspot features selected from those in the T3-FS4 feature set, as shown in Table 6.11.

Flow Results, Best Combination of T3-FS4 Feature Weights					
	712R02	819R09	819R10	819R32	819R64
Xconfig. hotspot contrib.	-10.23	4.09	3.91	0.17	1.72
Xconfig. sin hotspot dir. * hotspot contrib.	6.57	-1.46	-3.35	-0.03	3.42
Xconfig. cos hotspot dir. * hotspot contrib.	16.44	-10.04	-0.27	1.90	-0.39
Oconfig. hotspot contrib.	0.40	-4.77	-0.58	0.34	-0.39
Oconfig. sin hotspot dir. * hotspot contrib.	-5.60	-7.34	2.10	0.35	-1.63
Oconfig. cos hotspot dir. * hotspot contrib.	-0.84	26.14	-1.31	-1.10	1.04
sin seg. 1 dir. * seg. 1 contrib.	-	-0.88	2.03	-0.54	-0.45
cos seg. 1 dir. * seg. 1 contrib.	-11.05	7.14	1.18	-	2.93
sin seg. 2 dir. * seg. 2 contrib.	-6.14	18.58	1.85	-	+
cos seg. 2 dir. * seg. 2 contrib.	6.90	-8.22	-	-1.60	+
sin seg. 3 dir. * seg. 3 contrib.	-8.18	-	-0.57	-	12.94
cos seg. 3 dir. * seg. 3 contrib.	-	-26.23	-	-4.11	-14.38
sin seg. 4 dir. * seg. 4 contrib.	-	-	0.66	-2.77	0.52
cos seg. 4 dir. * seg. 4 contrib.	3.07	2.04	-0.61	1.43	-
sin seg. 5 dir. * seg. 5 contrib.	2.67	0.71	-	+	+
cos seg. 5 dir. * seg. 5 contrib.	-	-	-	+	+
sin seg. 6 dir. * seg. 6 contrib.	-	-20.86	-	-	-
cos seg. 6 dir. * seg. 6 contrib.	-0.91	-	1.01	3.57	-15.16
sin seg. 7 dir. * seg. 7 contrib.	+	+	+	-	-
cos seg. 7 dir. * seg. 7 contrib.	+	+	+	3.36	7.13
sin seg. 8 dir. * seg. 8 contrib.	+	+	+	+	-
cos seg. 8 dir. * seg. 8 contrib.	+	+	+	+	1.70
sin seg. 9 dir. * seg. 9 contrib.	3.04	4.88	-0.01	1.75	-4.26
cos seg. 9 dir. * seg. 9 contrib.	12.67	-	-0.88	-2.78	-

Table 6.11: Best combination of T3-FS4 feature weights for LR flow data prediction. Refer to Figure 3.1(b) for segment numbering scheme. - denotes features not selected. + denotes features associated with hotspot locations and thus were not available for non-hotspot feature selection. Note that the hotspots are not the same for all neurons.

It is interesting how exactly 9 non-hotspot features were selected for each neuron although there was no restriction on the number of features that could be selected. This behavior may be due to over-fitting that occurs when more than 9 non-hotspot features are selected, thereby producing poor validation results, so those combinations of features were not selected. Note that the features that were selected are highly dependent on the structure of the 16 flow stimuli used for training and validation. The largest weights assigned to hotspot features were for neurons 712R02, 819R09, and 819R10, while the largest weights assigned to non-hotspot features were for neurons 819R32 and 819R64. The significance of a feature cannot necessarily be determined based on the relative magnitude of its weight. Low weights may be assigned to features with high pre-weighted values and vice versa. Pre-weighted values were not normalized for each feature category because doing so would remove inherent relationships between firing rate response values across the various segments. Preliminary testing showed that doing so produces models which perform poorly.

The most common non-hotspot feature selected for each neuron is the horizontal response contribution of segment 9, which is the upper rightmost segment. The least common non-hotspot featured selected for each neuron is the vertical response contribution for segment 6, which is located directly underneath segment 9. It is difficult to draw any reasonable conclusions from these results which apply to all of the neurons. This is understandable considering that each neuron may have a different physiological purpose and behavior.



## 6.5 Flow Data Linear Model Feature Weight Analysis

Each neuron's flow data linear model consists of weighted hotspot and non-hotspot features. In order to better understand the interaction effect between hotspot and non-hotspot segments we analyze the correlations among these features. For each neuron we consider the 20 top performing T3-FS4 linear models in terms of MSE. Each T3-FS4 model consists of different combinations of T3-FS4 features. By considering the top 20 combinations of features we can better understand the correlations between features than if we were to analyze only the best performing combination. This is because the most prominent correlations between pairs of features may exist regardless if a few other features are included in, or excluded from, the feature set combination. The more times the same feature pairing appears among the top performing feature set combinations, the more significant the feature pairing, and the less likely that the correlation was forced by the LR technique in order to model the data. Therefore, the correlations of the most significant feature pairings are the most likely to represent the underlying physiological behavior of the neuron and structure of its RF space.

Figure 6.10 shows the hotspot feature weights assigned to the top 20 combinations of T3-FS4 features for neuron 712R02. The leftmost bar of each feature group represents the feature weight of the best performing combination and rightmost bar represents the feature weight of worst performing combination among the 20, with the bars in between representing the feature weights of combinations of decreasing performance from left to right. Notice that

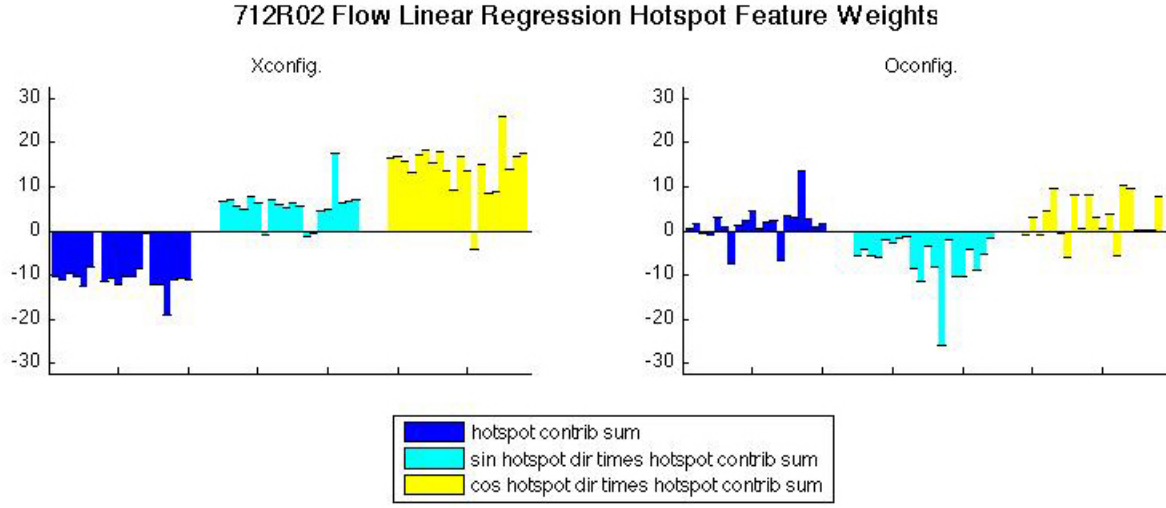


Figure 6.10: Neuron 712R02 LR flow prediction hotspot feature weights for best T3-FS4 combinations. Bars represent the feature weights of 20 combinations of decreasing performance from left to right for each feature type.

the majority of the combinations share a similar weight for each of the 3 X-configuration hotspot feature weights. This implies that each feature weight is significant and not a forced product of the LR technique. On the other hand, some of the non-hotspot feature weights shown in Figure 6.11 vary in non-trivial ways among the T3-FS4 combinations. Such noise may imply that such features are not as significant. Thus, high significance is related to low standard deviation between feature weights. In turn low standard deviation should increase the correlation value calculated between any two pairs of features. This is all taken into account when calculating the correlation coefficient using Equation 6.1.

### 712R02 Flow Linear Regression Non-Hotspot Feature Weights

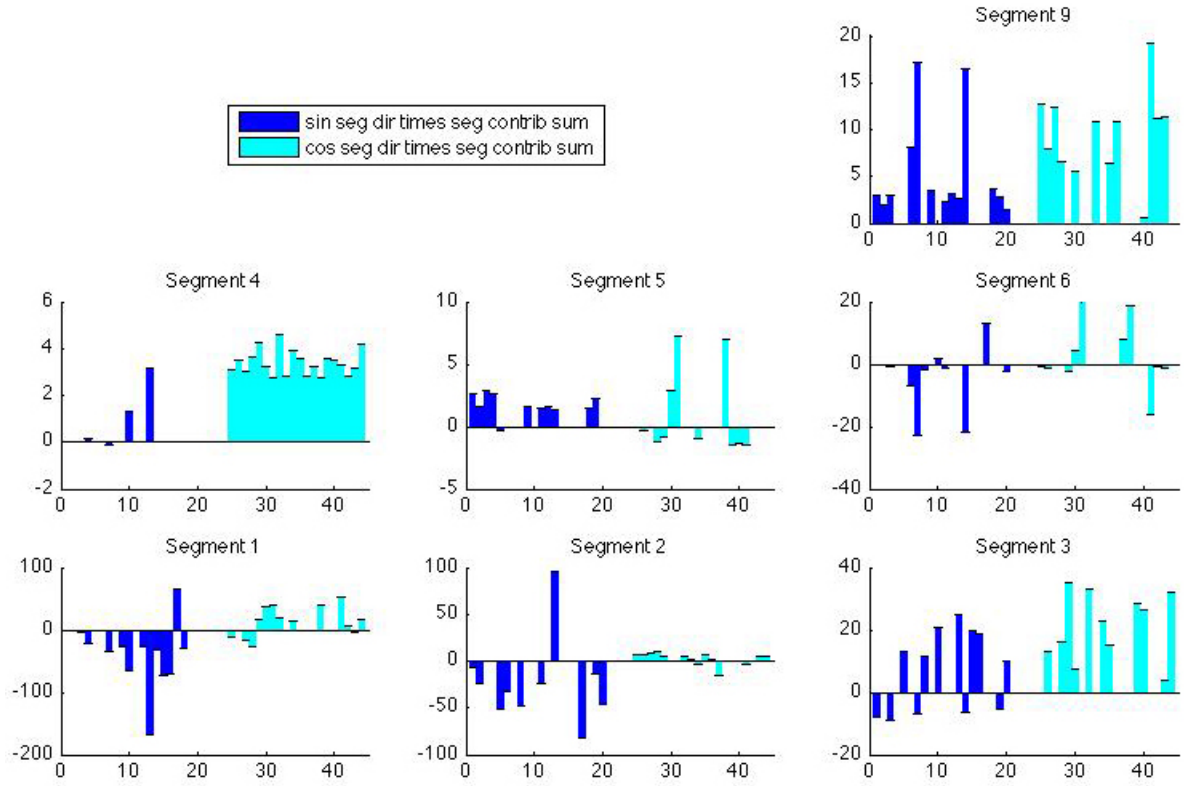


Figure 6.11: Neuron 712R02 LR flow prediction non-hotspot feature weights for best T3-FS4 combinations. Bars represent the feature weights of 20 combinations of decreasing performance from left to right for each feature type.

$$\rho_{x,y} = \frac{cov(X,Y)}{\sigma_x, \sigma_y} \quad (6.1)$$

Pearson's linear correlation coefficient,  $\rho$ , between feature  $x$  and  $y$  is calculated by dividing the covariance between those features by the product of their individual standard deviations.  $X$  is the set consisting of all the feature  $x$  weights for all 20 of the top performing combinations and is used to calculate the covariance between feature  $x$  and  $y$  as well as  $\sigma_x$ , the standard deviation for feature  $x$ . Similarly,  $Y$  is the set consisting of all the feature  $y$  weights for all 20 of the top performing combinations and is used to calculate both the covariance between feature  $x$  and  $y$  as well as  $\sigma_y$ , the standard deviation for feature  $y$ . A two-tailed test was performed to test the hypothesis of no correlation against the alternative that there is a non-zero correlation. If the calculated significance value, known as the p-value, is less than 0.05 then the associated correlation coefficient is significantly different from zero. Therefore, lower p-values indicate a stronger probability of correlation between feature weights. Correlation metrics for neuron 712R02 are shown in Figure 6.12 where the index assigned to each feature is shown in Table 6.12. Note that the places corresponding to non-hotspot features 19 through 22 are left blank in each of the plots in Figure 6.12 because segments 7 and 8 correspond to the X-configuration and O-configuration hotspot segments, respectively. The 20 top performing combinations of feature weights used for linear model feature weight analysis are listed in Appendix H.

T3-FS4 Feature Indices	
1	Xconfig. hotspot contrib.
2	Xconfig. sin hotspot dir. * hotspot contrib.
3	Xconfig. cos hotspot dir. * hotspot contrib.
4	Oconfig. hotspot contrib.
5	Oconfig. sin hotspot dir. * hotspot contrib.
6	Oconfig. cos hotspot dir. * hotspot contrib.
7	sin segment 1 dir. * segment 1 contrib.
8	cos segment 1 dir. * segment 1 contrib.
9	sin segment 2 dir. * segment 2 contrib.
10	cos segment 2 dir. * segment 2 contrib.
11	sin segment 3 dir. * segment 3 contrib.
12	cos segment 3 dir. * segment 3 contrib.
13	sin segment 4 dir. * segment 4 contrib.
14	cos segment 4 dir. * segment 4 contrib.
15	sin segment 5 dir. * segment 5 contrib.
16	cos segment 5 dir. * segment 5 contrib.
17	sin segment 6 dir. * segment 6 contrib.
18	cos segment 6 dir. * segment 6 contrib.
19	sin segment 7 dir. * segment 7 contrib.
20	cos segment 7 dir. * segment 7 contrib.
21	sin segment 8 dir. * segment 8 contrib.
22	cos segment 8 dir. * segment 8 contrib.
23	sin segment 9 dir. * segment 9 contrib.
24	cos segment 9 dir. * segment 9 contrib.

Table 6.12: T3-FS4 feature indices.

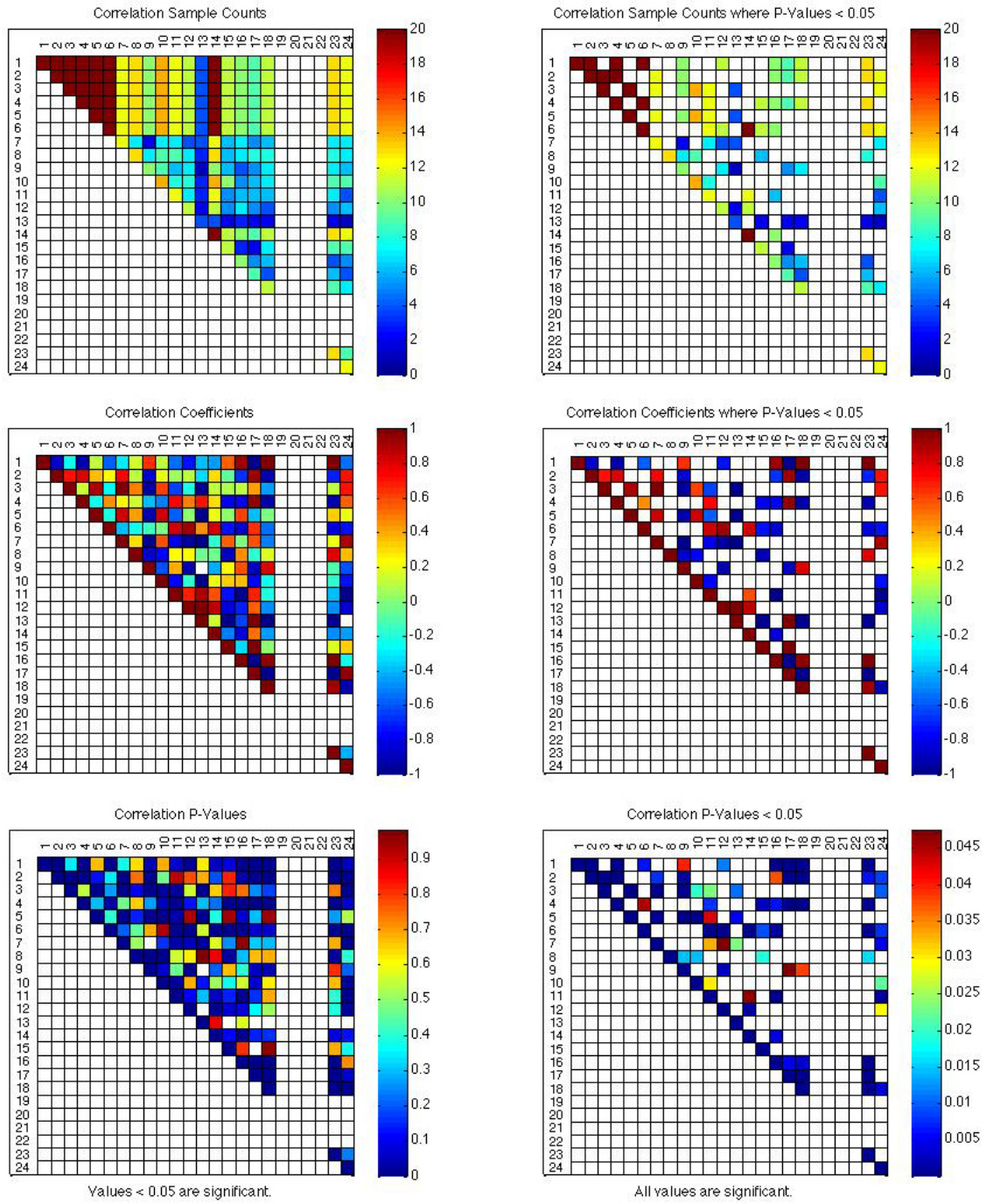


Figure 6.12: Neuron 712R02 LR flow prediction feature weight correlations for best T3-FS4 combinations. Top plots show the number of times each feature pairing appeared among the top combinations. Center plots show the correlation coefficient values. Bottom plots show the associated p-value for each correlation coefficient. Right plots only show metrics for p-values < 0.05.

712R02 Flow Hotspot Non-Hotspot Feature Correlation

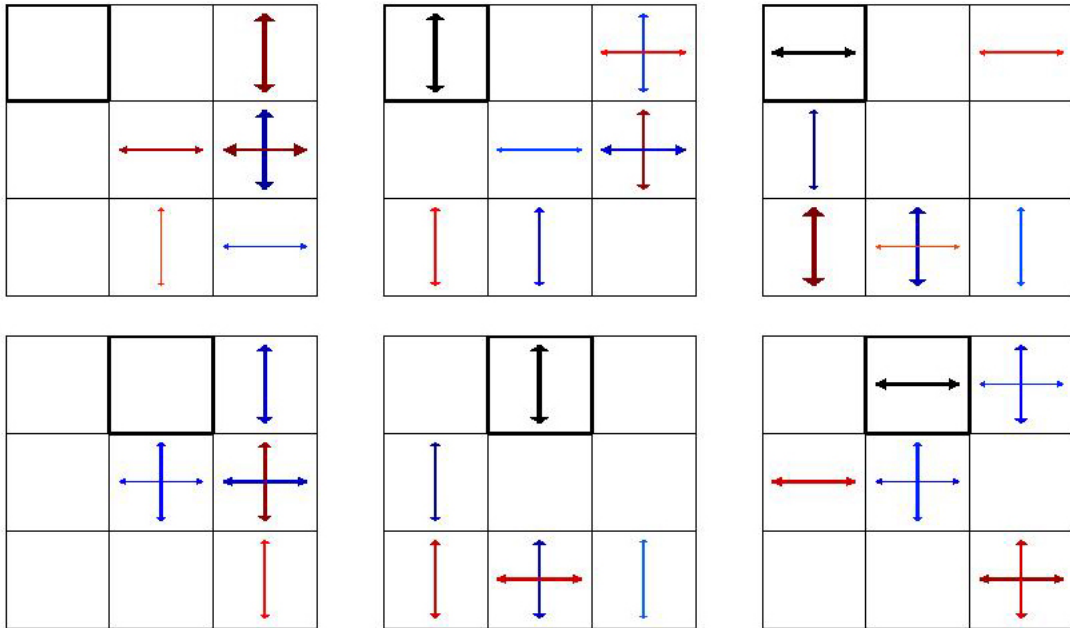


Figure 6.13: Neuron 712R02 LR flow prediction feature weight correlation visualization for best T3-FS4 combinations. Top plots show correlations with X-configuration hotspot features. Bottom plots show correlations with O-configuration hotspot features. See description in text.

A visualization of correlations between pairs of hotspot and non-hotspot features is shown in Figure 6.13. The hotspot segment for each feature pairing is outlined in black. No arrow inside of the black outline indicates the feature corresponding to the hotspot contribution for that segment. A black vertical arrow inside of the black outline indicates the feature corresponding to the sine of the hotspot direction times the hotspot contribution for that segment. A black horizontal arrow inside of the black outline indicates the feature corresponding to the cosine of the hotspot direction times the hotspot contribution for that segment. Similarly, colored arrows inside of segments without black outlines correspond to non-hotspot features for those segments, where the orientation of the arrow indicates the sine and cosine features in the same way a black arrow indicates those features for hotspot segments.

Each colored arrow (i.e. each non-hotspot segment feature) shown is paired with the black arrow (i.e. the hotspot segment feature) shown within the corresponding 9-segment plot. The color of the non-hotspot segment arrow corresponds to the correlation coefficient value for that pairing according to the correlation coefficient color scale shown in Figure 6.12, where red represents positive correlation and blue represents negative correlation. Only correlations with p-values  $< 0.05$  are shown. The thickness of each arrow represents the p-value associated with the correlation coefficient. The thicker the arrow the smaller the p-value and greater the significance. Figures 6.14 and 6.15 show the correlation metrics and visualization for neuron 819R64. Figures G.1 through G.6 in Appendix G show correlation information for other neurons.



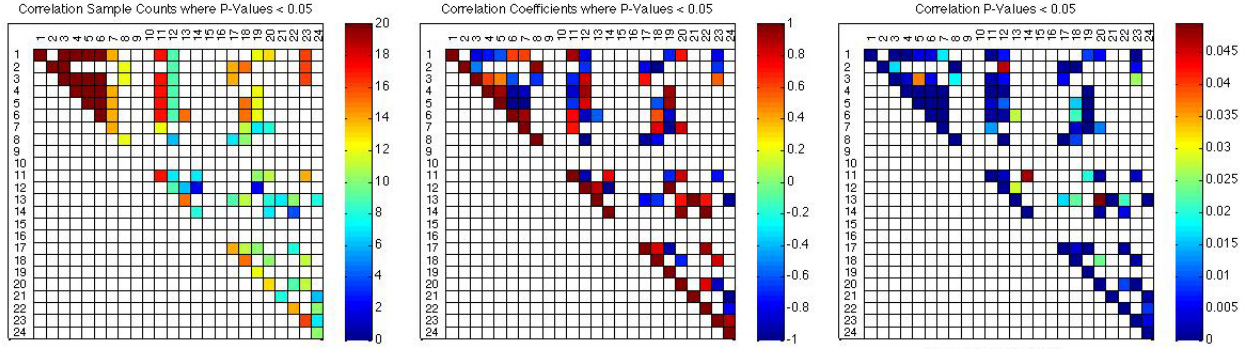


Figure 6.14: Neuron 819R64 LR flow prediction feature weight correlations for best T3-FS4 combinations.

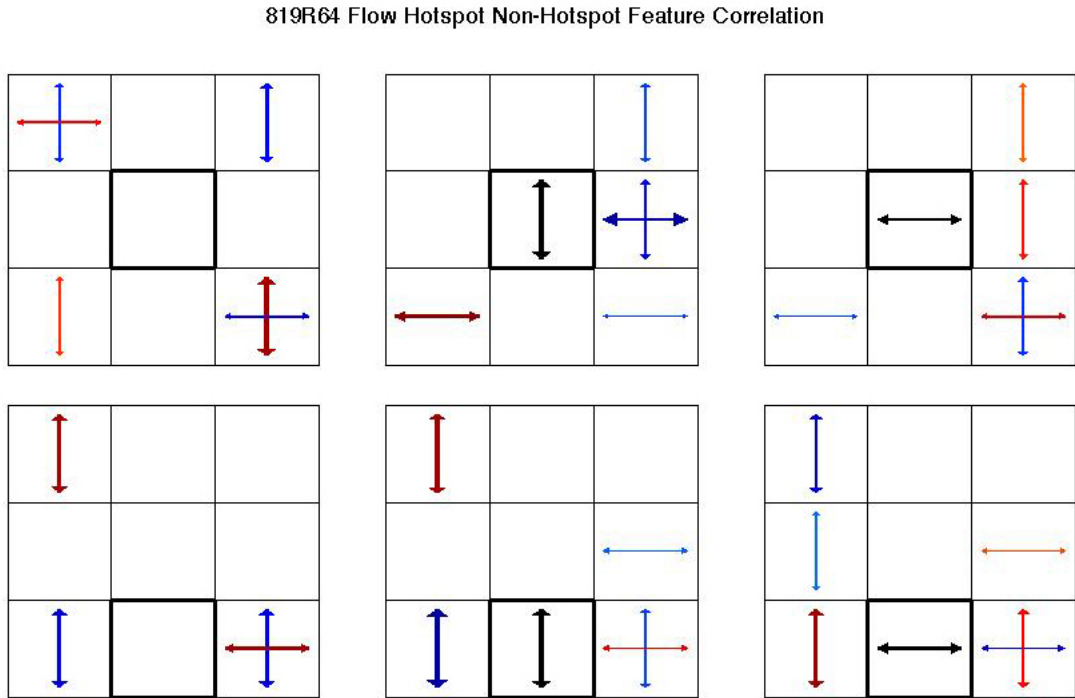


Figure 6.15: Neuron 819R64 LR flow prediction feature weight correlation visualization for best T3-FS4 combinations.

The correlation visualization for each neuron illustrates the interaction effect between that neuron’s hotspot and non-hotspot segments. The most significant correlations are those which are predominantly positively or negatively correlated, with the correlations falling in between the extremes having less significance. Overall, there is a lot of variation among the most significantly correlated feature types and their segment positions for each neuron. The bottom two rightmost plots in Figure 6.15 for Neuron 819R64 show that nearly all of the most significant feature correlations change polarity depending on the orientation of the hotspot feature. Specifically, negative correlations corresponding to vertical hotspot features become positive correlations for horizontal hotspot features, and vice versa. This applies to both the X-configuration hotspot correlations (upper two rightmost plots) and O-configuration hotspot correlations (bottom two rightmost plots) for neuron 819R64. We call this phenomenon feature correlation polarity inversion. Considering that neuron 819R64 is very strongly characterized by opponent organization, as seen in Figure 3.8, this behavior makes sense. Feature correlation polarity inversion is also exhibited by neuron 819R09 and 819R10 to a lesser degree.

## 6.6 Linear Regression Technique Applied to Doubles Data

Similar to how the LR technique was applied to flow data, the LR technique was also applied to doubles data. Again, all-inclusive, segmental, and exhaustive combinations of the feature sets shown in Table 5.3, 5.4, and 5.5 were provided as inputs to the LR technique for 5 neurons. Note that feature sets 2 and 3 were omitted because they include features calculated using the doubles pairing response, which is what we are trying to predict for doubles data. LOO-CV was performed on the doubles data by partitioning the 112 recorded data points into 112 partitions of equal size. LR was not calculated with a constant value because preliminary tests showed that the use of a constant value produced models that did not perform as well as those without a constant value. The LR LOO-CV results for doubles data are shown in Table 6.13, 6.14, and 6.15. Note that each table only shows the MSE and absolute error values for the best performing combination for each feature set. Plots of the best performing T3-FS4 combination for each neuron are shown in Figure 6.16 as well as in Figure I.1 through I.4 in Appendix I.

Based on these results the exhaustive combinations of features in sets T3-FS1 and T3-FS4 produced models that are slightly more capable of predicting firing rate responses for doubles data than combinations of features in other sets. Combinations of features in set T3-FS1 and T3-FS4 perform the best for 4 out of the 5 neurons in terms of MSE when compared to any other feature set. Table 6.16 and 6.17 show the hotspot locations and feature weights for the best combination of T3-FS1 features and T3-FS4 features, respectively.

Doubles Results, Combination of All Features in Set										
	712R02 Test		819R09 Test		819R10 Test		819R32 Test		819R64 Test	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
T1-FS1	36.89	4.89	30.63	3.53	1430.20	28.04	20.21	3.26	10.95	2.50
T1-FS2	-	-	-	-	-	-	-	-	-	-
T1-FS3	-	-	-	-	-	-	-	-	-	-
T1-FS4	44.44	5.23	30.46	3.43	1504.03	30.47	24.09	3.56	12.94	2.76
T1-FS5	30.33	4.59	31.47	3.43	1611.23	31.91	25.18	3.68	14.07	2.90
T2-FS1	42.48	5.20	36.19	4.03	1435.63	28.06	31.65	4.42	14.73	2.95
T1-FS2	-	-	-	-	-	-	-	-	-	-
T1-FS3	-	-	-	-	-	-	-	-	-	-
T2-FS4	56.57	6.13	36.04	3.96	1520.16	30.90	46.78	5.69	18.37	3.45
T2-FS5	52.63	5.92	37.25	3.95	1626.54	31.48	39.41	4.96	17.19	3.33
T3-FS1	19.69	3.26	30.08	3.44	1392.38	28.42	16.35	2.95	8.09	2.22
T1-FS2	-	-	-	-	-	-	-	-	-	-
T1-FS3	-	-	-	-	-	-	-	-	-	-
T3-FS4	29.96	3.89	30.81	3.46	1483.31	30.13	20.03	3.29	9.49	2.39
T3-FS5	21.60	3.57	31.81	3.55	1599.51	31.21	20.08	3.31	10.17	2.48
MaxFiringRate	41.25		44.67		164.00		33.80		22.00	

Table 6.13: LR doubles prediction results using combination of all set features.

Doubles Results, Best Segmental Combination of Features in Set										
	712R02 Test		819R09 Test		819R10 Test		819R32 Test		819R64 Test	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
T1-FS1	35.70	4.76	27.31	3.20	1318.81	27.35	19.04	3.14	10.20	2.37
T1-FS2	-	-	-	-	-	-	-	-	-	-
T1-FS3	-	-	-	-	-	-	-	-	-	-
T1-FS4	43.47	5.13	27.84	3.17	1293.17	27.52	19.70	3.25	11.01	2.44
T1-FS5	29.27	4.44	27.48	3.11	1334.17	27.98	19.85	3.28	11.06	2.44
T2-FS1	39.14	5.03	30.70	3.91	1328.31	27.25	31.37	4.33	13.47	2.82
T1-FS2	-	-	-	-	-	-	-	-	-	-
T1-FS3	-	-	-	-	-	-	-	-	-	-
T2-FS4	47.23	5.54	29.81	3.77	1297.31	27.67	37.26	4.98	15.99	3.12
T2-FS5	40.47	5.07	29.32	3.61	1343.61	27.93	33.36	4.56	14.23	2.95
T3-FS1	18.64	3.21	26.41	3.18	1298.79	27.38	15.71	2.92	7.59	2.08
T1-FS2	-	-	-	-	-	-	-	-	-	-
T1-FS3	-	-	-	-	-	-	-	-	-	-
T3-FS4	26.65	3.59	26.96	3.28	1273.16	27.62	16.30	2.94	8.03	2.15
T3-FS5	19.49	3.33	26.43	3.17	1319.14	28.04	16.44	2.96	7.99	2.12
MaxFiringRate	41.25		44.67		164.00		33.80		22.00	

Table 6.14: LR doubles prediction results using best segmental combination of set features.

Doubles Results, Best Exhaustive Combination of Features in Set										
	712R02 Test		819R09 Test		819R10 Test		819R32 Test		819R64 Test	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
T1-FS1	35.70	4.76	27.31	3.20	1318.81	27.35	19.04	3.14	10.20	2.37
T3-FS2	-	-	-	-	-	-	-	-	-	-
T3-FS3	-	-	-	-	-	-	-	-	-	-
T1-FS4	43.22	5.08	26.73	3.07	1269.02	27.15	19.58	3.22	10.88	2.44
T3-FS5	-	-	-	-	-	-	-	-	-	-
T2-FS1	39.14	5.03	30.70	3.91	1328.31	27.25	31.37	4.33	13.47	2.82
T3-FS2	-	-	-	-	-	-	-	-	-	-
T3-FS3	-	-	-	-	-	-	-	-	-	-
T2-FS4	46.26	5.48	28.93	3.72	1278.99	27.34	36.90	4.95	15.82	3.10
T3-FS5	-	-	-	-	-	-	-	-	-	-
T3-FS1	18.64	3.21	26.41	3.18	1298.79	27.38	15.71	2.92	7.59	2.08
T3-FS2	-	-	-	-	-	-	-	-	-	-
T3-FS3	-	-	-	-	-	-	-	-	-	-
T3-FS4	25.50	3.55	25.79	3.18	1258.26	27.43	16.19	2.95	7.94	2.14
T3-FS5	-	-	-	-	-	-	-	-	-	-
MaxFiringRate	41.25		44.67		164.00		33.80		22.00	

Table 6.15: LR doubles prediction results using best exhaustive combination of set features.

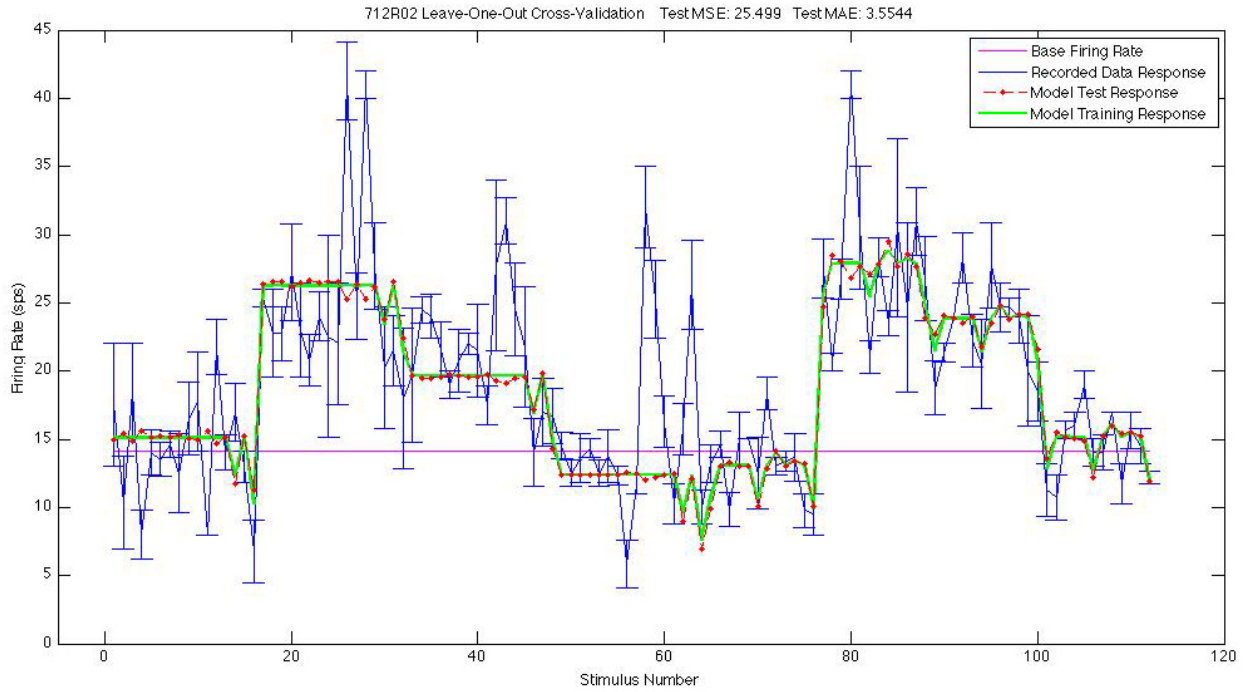


Figure 6.16: Neuron 712R02 LR doubles prediction using best T3-FS4 combination.

Doubles Results, Best Combination of T3-FS1 Feature Weights					
	712R02	819R09	819R10	819R32	819R64
Xconfig. hotspot contrib.	1.20	0.60	1.90	0.36	1.17
Xconfig. sin hotspot dir. * hotspot contrib.	-0.17	-0.04	-1.53	0.37	-0.97
Xconfig. cos hotspot dir. * hotspot contrib.	-1.08	0.00	2.12	0.35	-0.41
Oconfig. hotspot contrib.	1.16	1.05	0.38	1.29	0.96
Oconfig. sin hotspot dir. * hotspot contrib.	1.33	-0.84	-0.54	0.23	0.03
Oconfig. cos hotspot dir. * hotspot contrib.	-0.84	-0.60	-0.08	0.21	0.01
seg. 1 contrib.	-	-	-	-0.62	0.72
seg. 2 contrib.	-0.67	-	-	0.15	+
seg. 3 contrib.	-	-	-	-	-
seg. 4 contrib.	-	0.73	0.21	-	-
seg. 5 contrib.	0.50	1.17	-0.38	+	+
seg. 6 contrib.	0.37	-	-	-	-
seg. 7 contrib.	+	+	+	-	-
seg. 8 contrib.	+	+	+	+	0.71
seg. 9 contrib.	-	-	-	-0.29	-

Table 6.16: Best combination of T3-FS1 feature weights for LR doubles data prediction. - denotes features not selected. + denotes features associated with hotspot locations and thus were not available for non-hotspot feature selection. Note that the hotspots are not the same for all neurons.

Doubles Results, Best Combination of T3-FS4 Feature Weights					
	712R02	819R09	819R10	819R32	819R64
Xconfig. hotspot contrib.	1.24	0.46	1.75	0.46	1.34
Xconfig. sin hotspot dir. * hotspot contrib.	-0.01	-0.25	-1.32	0.37	-0.97
Xconfig. cos hotspot dir. * hotspot contrib.	-1.37	0.19	1.92	0.14	-0.54
Oconfig. hotspot contrib.	1.16	1.05	0.36	1.30	0.91
Oconfig. sin hotspot dir. * hotspot contrib.	1.32	-0.86	-0.54	0.18	0.00
Oconfig. cos hotspot dir. * hotspot contrib.	-0.83	-0.69	-0.05	0.23	0.05
sin seg. 1 dir. * seg. 1 contrib.	-	-	-	-	-
cos seg. 1 dir. * seg. 1 contrib.	-	-	-	-	0.72
sin seg. 2 dir. * seg. 2 contrib.	-	-	-	-	+
cos seg. 2 dir. * seg. 2 contrib.	-0.42	-	-	-	+
sin seg. 3 dir. * seg. 3 contrib.	-	1.00	-	0.36	-
cos seg. 3 dir. * seg. 3 contrib.	-	-	0.31	-	-
sin seg. 4 dir. * seg. 4 contrib.	0.25	-	-0.46	-	-
cos seg. 4 dir. * seg. 4 contrib.	-	0.79	-	-0.22	-
sin seg. 5 dir. * seg. 5 contrib.	-	-2.72	-	+	+
cos seg. 5 dir. * seg. 5 contrib.	-	-1.82	-	+	+
sin seg. 6 dir. * seg. 6 contrib.	-0.54	-	-	0.17	-
cos seg. 6 dir. * seg. 6 contrib.	-	-	-	-	-
sin seg. 7 dir. * seg. 7 contrib.	+	+	+	-	-
cos seg. 7 dir. * seg. 7 contrib.	+	+	+	-	-
sin seg. 8 dir. * seg. 8 contrib.	+	+	+	+	-
cos seg. 8 dir. * seg. 8 contrib.	+	+	+	+	-
sin seg. 9 dir. * seg. 9 contrib.	-0.90	-	-	-	-
cos seg. 9 dir. * seg. 9 contrib.	-	-	-	-	-6.23

Table 6.17: Best combination of T3-FS4 feature weights for LR doubles data prediction. - denotes features not selected. + denotes features associated with hotspot locations and thus were not available for non-hotspot feature selection. Note that the hotspots are not the same for all neurons.

The most common T3-FS1 non-hotspot feature selected for each neuron is the segment 5 response contribution, which makes sense considering that segment is directly in the center of the screen where the primate was directed to focus his attention. The T3-FS4 non-hotspot features selected for each neuron have very little in common, other than each neuron selected between 2 and 4 features each, which is significantly less than the 9 T3-FS4 non-hotspot features selected for each neuron's flow data linear model. The reason for this may be because less variation is present in the doubles data due to the absence of motion in 7 out of 9 segments per stimulus (since only the hotspot and one non-hotspot segment are active). A total of 8 or 9 segments are active per flow stimulus which provides a lot more variation between stimuli. This explanation is further supported by the fact that the linear model for doubles data is only capable of predicting a general response pattern across subsets of the doubles stimuli, as seen in Figure 6.16 and those in Appendix I, often lacking the precision to accurately predict spikes in the response values that deviate away from the average. In comparison, the presence of more variation between the flow stimuli allows for very precise predictions, as seen in Figure 6.9 and those in Appendix F.



## Chapter 7

### Discussion

#### 7.1 Physiological Realism

It is unknown how many RFs compose any given neuron in the MST region, their relative sizes and positions, and how they interact with each other. This thesis research provides support for the theory that RFs are sensitive to specific directions of motion and that each neuron is capable of reacting to multiple directions of motion since each neuron is associated with multiple RFs. When extending the concept of 2-D (x, y) RFs into higher feature spaces the time between when a subject perceives a stimulus and the neuronal response evoked is typically chosen as the third dimension, as exemplified by the Gaussian derivative model in Section 2.3. This implies a RF can behave differently depending on the length of time a stimulus is being perceived and how that stimulus changes over time. In this thesis we are more concerned with modeling average RF responses to consistent stimulus motion over a short period of time and not how the behavior of RFs change with respect to changes in stimulus motion over that time period. We extend the concept of 2-D RFs into the third dimension by modeling each RF's directional sensitivity, which is a less explored idea. For the purpose of simplification, we treat each RF as a static entity that is time-invariant; however, our model can possibly be

extended into the fourth dimension to account for changes over time as well.

We were able to accurately model and predict neuron firing rate behavior for doubles stimuli by subdividing the RF feature space into subspaces based on the direction of motion through hotspot regions. This implies that each hotspot acts like a switch capable of changing the general behavior of the neuron. The switch is triggered by the direction of stimulus motion across it and responds by enabling neuron inputs from certain RFs while disabling inputs from others. One issue with the switching theory is that only RFs corresponding to one of 8 subspaces are enabled at any given time. Since Gaussians are not shared between subspaces, many Gaussians must be trained to model all of the doubles data. Our results show that the hierarchical model for each neuron was composed of between 27 and 31 Gaussians. That number of RFs is not unrealistic considering that there are about one million optic nerves that converge into potentially hundreds of thousands of RFs to send information to the visual cortex. It is unknown how many RFs send signals to each individual neuron.

RF signals from the retina pass through a series of regions before arriving at the MST region, for example the V1, V2, and MT regions. Each of these regions integrates information from multiple RF sources to process color and directional information, detect edges, etc., before sending signals to the MST region. Therefore, many of the signals sent to the MST region from previous regions have already integrated the signals from many RFs. Clearly the operation of MST neurons is non-trivial and there are multiple steps to

detecting self-motion, performed by each of the intermediary regions between the retinal RFs and the MST region, that this thesis attempts to abstract away. At best, the models presented in this thesis are a vast simplification of the physiological processes of the mammalian visual system. Even so, we made a faithful attempt to simplify our models as much as possible in order to reduce unnecessary complexity and further our understanding of the basic operation of MST neurons.

Physiologically speaking, it makes more sense to assume that instead of any given RF being enabled or disabled, each RF is always active and produces a firing rate contribution of some kind. After all, parts of the brain do not completely shut off when they aren't in use. Thus, instead of assuming the discrete case, where any given RF is either producing a signal or not, a more appropriate model may assume a more continuous case, where each RF is producing a signal that is potentially modulated to become a partial signal. The latter case assumes that a given RF's firing rate contribution is initially the same regardless of the direction of motion through hotspot regions. Hotspot regions would then modulate the firing rate contributions of those RFs depending on the direction of motion through those hotspots in relation to the direction of motion through non-hotspot regions.

Yu applied static gain factors to a DGM trained on singles or doubles data in an attempt to model flow data. The gain factor values were determined using a GA trained on flow data. Alternatively, one could apply dynamic gain factors to a SGMM trained on singles data in an attempt to model doubles

and flow data. The function used to calculate the gain factor values would be determined by inspecting doubles and flow data. Unlike static gain factors, which are multiplicative constants, dynamic gain factors would change based on differences between the direction of stimulus motion through hotspot and non-hotspot regions. This would allow singles, doubles, and flow stimuli responses to be modeled following the same procedure using the same common set of RFs, which is desirable. The linear model presented in this thesis assumes that there is an extra step involved when determining responses to flow stimuli because it must first extract features from the SGMM before combining them into a single response. This step is not involved when responding to singles or doubles stimuli, which determine responses from the SGMM(s) directly. It is entirely possible that the intermediary regions between the retinal RFs and the MST region of the visual system perform the same kind of feature extraction and provide those feature input signals to the MST region, although it is unlikely that would occur only for flow stimuli. It is more likely that the visual system has no notion of the difference between singles, doubles, and flow stimuli and responds to all of them following the same procedure using signals from the same set of RFs.

## **7.2 Dynamic Gain Theory**

By analyzing the correlations between feature weights employed by our LR technique it is obvious that the relationship between the effect of the direction of motion through hotspot regions and the direction of motion

through non-hotspot regions on segment firing rate response contributions is non-trivial. Once these correlations have been determined one can use that information to determine how to calculate dynamic gain factors. For any given RF, dynamic gain factors can be calculated as a function of the direction of motion through the hotspot regions and the direction of motion through the non-hotspot segment(s) in which the RF is positioned. The fact that flow data can be accurately modeled using features that multiply trigonometric functions calculated using directions of motion through stimulus regions by SGMM segment firing rate responses (i.e.  $(\sin(seg. dir.) * seg. contrib.)$ , and  $(\cos(seg. dir.) * seg. contrib.)$ ) offers strong support for the use of dynamic gain factors. This is because the non-trigonometric factor of those features is based on a static structure of a common set of underlying RFs in the SGMM and the trigonometric factor is based on the direction of motion through a hotspot or non-hotspot stimulus region.

In the physiological sense, dynamic gain factors can be explained by the simple divisive modulation of one RF response by another RF response. We assume that pre-MST regions of the visual system generate at least two types of signals passed to the MST region by: a) responding to stimuli in specific regions of the visual field using the type of SGMM RFs presented in this thesis, and b) responding to differences in the direction of motion through hotspot and non-hotspot regions of the visual field. The latter signals modulate the former signals resulting in the application of dynamic gain factors, as illustrated in Figure 7.1. Divisive modulation can only decrease an existing the firing rate

response, thereby resulting in a partial response. We assume that the strength of modulation is related to the magnitude of the difference between directions of motion through hotspot and non-hotspot regions. Gain factors may also be applied when no hotspot motion is present, which may result in a default reduction in the firing rate response of RFs corresponding to certain regions of the visual field.

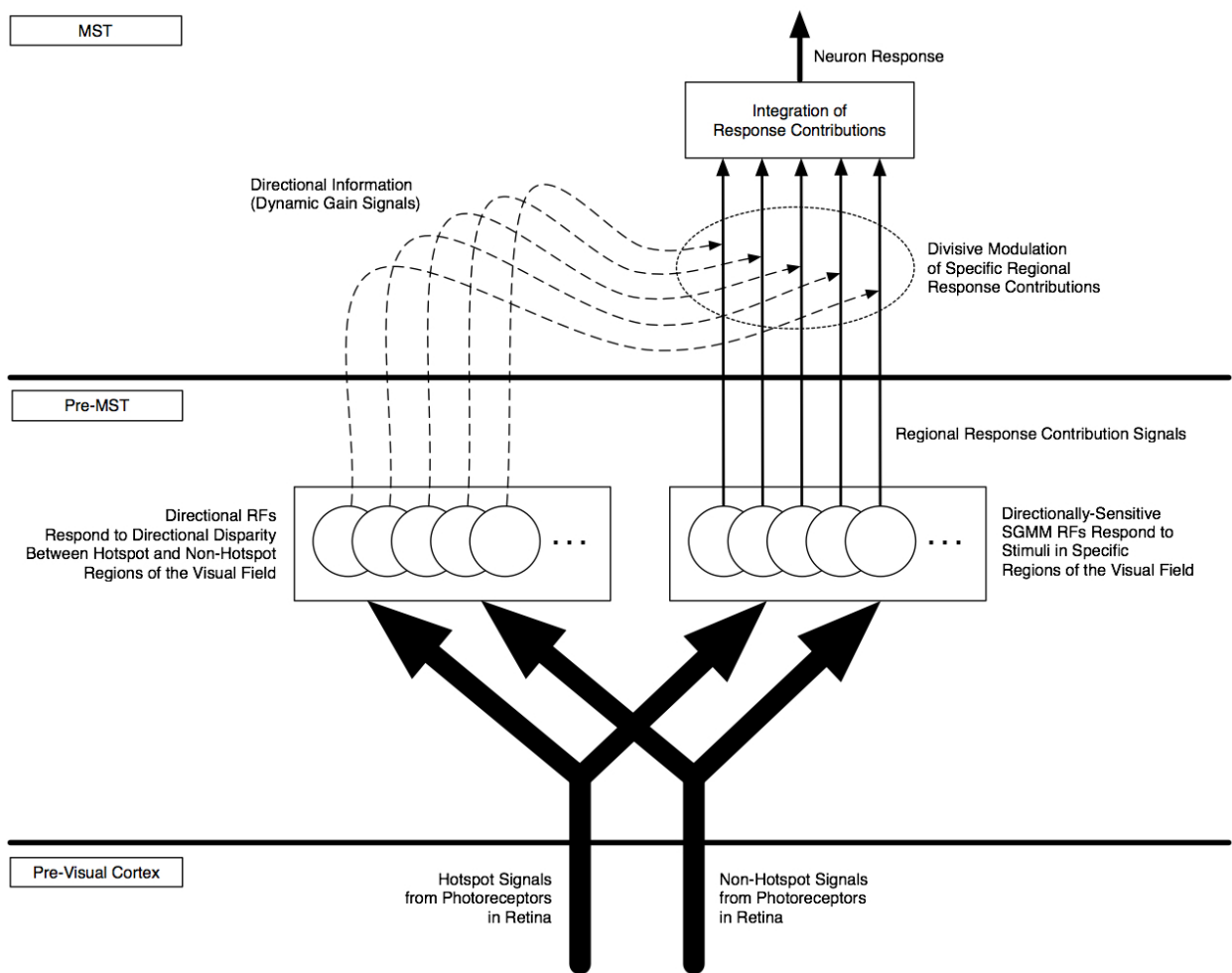


Figure 7.1: Physiological production of MST neuron response using dynamic gain factors. See description in text.

## Chapter 8

## Conclusion

### 8.1 Summary

This thesis presents a linear regression (LR) technique applied to non-linear features extracted from a scaled Gaussian mixture model (SGMM) to describe the receptive field (RF) behavior of neurons in the Medial Superior Temporal (MST) region of the brain of Rhesus monkeys which viewed visual stimuli on a projector screen. The stimuli consist of simple and complex combinations of planar, circular, and radial motion simulating the effects of self-motion. The SGMM is capable of effectively modeling the primate's neuronal responses to singles stimuli and predicting how the RFs in the primate's neurons respond to certain novel singles stimuli, which provides support for Duffy and Wurtz's overlapping gradients hypothesis. We train the SGMM using a genetic algorithm (GA) and successfully generate multivariate Gaussian distributions to fit firing rate response values by creating species of Gaussians with similar properties. Each species is trained independently of the others in order to preserve innovation and not prematurely discard potential solutions. The GA competes against a custom greedy algorithm employing Gaussians of known shapes, sizes, and positions to ensure that the SGMM's residual error is reduced by the addition of each new Gaussian during each stage of training.



Results are summarized by neuron in Appendix J.

We have shown that the primate’s neuronal responses to doubles stimuli are influenced by the direction of motion within the most active regions in the primate’s RF space. This is known as the interaction effect and is supported by our discussion of opponent organization. By training a separate SGMM corresponding to each cardinal direction of motion within each hotspot segment we effectively model the primate’s neuronal responses to doubles stimuli and predict how the RFs in the primate’s neurons respond to certain novel doubles stimuli. We have also provided evidence that SGMMs which do not restrict Gaussian alignment to one direction of motion perform better than those which enforce such a restriction. Furthermore, we have provided evidence that SGMMs trained using the cumulative distribution function (CDF) instead of the probability distribution function (PDF) perform better for certain partitions of training and test data for the same neuron.

After extracting non-linear features from the SGMM trained on singles data, the LR technique is able to train a linear model for each neuron that is capable of modeling and predicting flow data responses for all flow stimuli with remarkable precision and accuracy. On the other hand, the linear model trained by the LR technique to model and predict doubles data is not as precise and is only capable of modeling and predicting the average response values for the various doubles stimuli for each neuron. This difference is most likely caused by the lack of variation between doubles stimuli compared to the variation between flow stimuli, which is a consequence of fewer active

segments. The LR technique is able to produce the best performing linear models for each of the neurons using the same set of hotspot and non-hotspot features, which includes magnitudinal hotspot features, directional hotspot features, and directional non-hotspot features. This provides support for our hypothesis that the non-linear interaction between RFs can be explained by the relationship between hotspot and non-hotspot segments and how that relationship is influenced by the direction of motion within hotspot segments.

Through our analysis of weights assigned to features by the LR technique we were able to determine the most significant interactions between features by calculating their correlation coefficients. We showed that features corresponding to the neuron which most strongly exhibited opponent organization behavior for doubles data reflected the directional sensitivity of that neuron. Specifically, a  $180^\circ$  change in the direction of motion within the hotspot segment of that neuron would cause an inhibitory response to become an excitatory response, and vice versa. The correlations between features calculated for the that neuron reverse polarity based on the vertical or horizontal orientation of the hotspot segment feature. This result provides support for the opponent organization theory presented by Steinmetz, et al., as well as our hypothesis.

## 8.2 Future Work

This research offers many opportunities for future work. The main advantage of using the CDF over the PDF when training SGMMs is that the CDF performs integration over a volume of the RF space and therefore is sensitive to stimulus size. On the other hand, the PDF is calculated at infinitesimally small points in the RF space and is not sensitive to the stimulus size. It was acceptable to use the PDF in our experiments because each stimulus occupied 1 of 9 equally-sized segments, so stimulus size was not a factor. It would be interesting to see how accurately a SGMM trained using the CDF is capable of modeling and predicting firing rate response values for stimuli consisting of various sizes. Through our experimentation we have discovered that the integration calculations required to employ the CDF SGMM are quite computationally expensive. It is simply too time-consuming to run such calculations on an average personal computer even when the code is optimized to run in C++. Part of the problem is related to the overhead required to make C++ function calls from Matlab using the MEX interface. One alternative would be to implement the entire algorithm in C++ or another non-interpreted programming language. Alternatively, the Matlab program can be run on a super computer or grid computing system.

Another opportunity for future work is to create a universal model that is capable of modeling and predicting singles, doubles, and flow stimuli responses for the same neuron. In Section 7 we presented the idea of using dynamic gain factors for this purpose. The experiments performed in this

thesis use one SGMM to model singles data, 8 SGMMs to model separate partitions of doubles data, and one linear model for flow data (or doubles data) for the same neuron. We essentially treat the singles data and flow data as different data types because the flow data linear model performs feature extraction on a singles data SGMM, which implies that the flow data requires higher-level processing to model than the singles data. We also treat the 8 partitions of doubles data as different data types by training a SGMM for each partition in isolation of the others. In reality the singles, doubles, and flow data are all of the same data type because we believe that the visual system does not make a special exception to respond to singles, doubles, and flow stimuli differently. The structure of the RF does not change depending in the type of stimuli being perceived, but it is possible that certain areas of the RF are more active or inactive for certain types of stimuli. We found it necessary to handle doubles and flow data differently from singles data to account for the regional interaction effect using our methodology. We desire a methodology that does not make a distinction between complex data and singles data. Such a model would not require explicit knowledge of the number of active stimulus segments in order to train a model capable of predicting responses.

One of the major reasons why we found it difficult to formulate such a methodology and universal model is lack of data. Specifically, the singles data corresponds to stimuli with 1 active segment, the doubles data corresponds to stimuli with 2 active segments, and the flow data corresponds to stimuli with 8 or 9 active segments. The jump in the number of active segments between the

doubles stimuli and flow stimuli is non-trivial. In the experiments performed by Duffy and Page the doubles stimuli cover  $22.\bar{2}\%$  of the subject's central visual field ( $90^\circ \times 90^\circ$ ) and the flow stimuli cover  $88.\bar{8}\%$  to 100%, which is a considerable difference. It would be desirable if in the future experiments are performed to collect data corresponding to 3, 4, 5, 6, or 7 active segments. Of course the data corresponding to 1, 2, 8, and 9 active segments would also need to be recorded for the same primate in relatively the same period of time in order to form a consistent and complete data set. Given the data available to us, we were able to grasp the interaction between hotspot and non-hotspot segments, which we believe to be the most important interaction, but not the only one. There may also be interactions between hotspot segments and multiple non-hotspot segments such that the relationship between the direction of motion within the non-hotspot segments can be used for modeling and prediction. It would be interesting to observe the change in firing rate response that occurs when another non-hotspot segment is activated for a doubles stimulus with a known firing rate response.

## Appendices

## Appendix A

### Minimum, Maximum, and Base Firing Rates by Neuron

Minimum, Maximum, and Base Firing Rates									
	Singles			Doubles			Flow		
	Min	Max	Base	Min	Max	Base	Min	Max	Base
712R02	4.20	40.25	14.13	5.80	41.25	14.13	5.75	101.67	42.00
819R09	0.80	17.40	4.27	0.00	44.67	4.27	0.00	88.80	19.67
819R10	0.00	128.33	24.77	0.00	164.00	24.77	5.00	91.50	3.00
819R32	0.80	21.00	10.06	0.00	33.80	10.06	0.40	39.20	9.50
819R64	0.80	16.50	2.92	0.33	22.00	2.92	0.40	32.25	3.00

Table A.1: Minimum, maximum, and base firing rates by neuron and stimulus type.

## Appendix B

### Singles Stimuli Numbering Scheme

Singles Oconfig. Stimuli Numbering Scheme		
Index	Direction	Segment
1	0°	2
2	0°	4
3	0°	6
4	0°	8
5	90°	2
6	90°	4
7	90°	6
8	90°	8
9	180°	2
10	180°	4
11	180°	6
12	180°	8
13	270°	2
14	270°	4
15	270°	6
16	270°	8

Table B.1: Singles Oconfig. stimuli numbering scheme.

Singles Xconfig. Stimuli Numbering Scheme		
Index	Direction	Segment
17	0°	1
18	0°	3
19	0°	5
20	0°	7
21	0°	9
22	90°	1
23	90°	3
24	90°	5
25	90°	7
26	90°	9
27	180°	1
28	180°	3
29	180°	5
30	180°	7
31	180°	9
32	270°	1
33	270°	3
34	270°	5
35	270°	7
36	270°	9

Table B.2: Singles Xconfig. stimuli numbering scheme.



# Appendix C

## Doubles Stimuli Numbering Scheme

Doubles Xconfig. Stimuli Numbering Scheme Part 1												
			HotspotSeg					NonHotspotSeg				
Index	HotspotDir	NonHotspotDir	712R02	819R09	819R10	819R32	819R64	712R02	819R09	819R10	819R32	819R64
1	0°	0°	7	7	7	5	5	1	1	1	1	1
2	0°	90°	7	7	7	5	5	1	1	1	1	1
3	0°	180°	7	7	7	5	5	1	1	1	1	1
4	0°	270°	7	7	7	5	5	1	1	1	1	1
5	0°	0°	7	7	7	5	5	3	3	3	3	3
6	0°	90°	7	7	7	5	5	3	3	3	3	3
7	0°	180°	7	7	7	5	5	3	3	3	3	3
8	0°	270°	7	7	7	5	5	3	3	3	3	3
9	0°	0°	7	7	7	5	5	5	5	5	7	7
10	0°	90°	7	7	7	5	5	5	5	5	7	7
11	0°	180°	7	7	7	5	5	5	5	5	7	7
12	0°	270°	7	7	7	5	5	5	5	5	7	7
13	0°	0°	7	7	7	5	5	9	9	9	7	7
14	0°	90°	7	7	7	5	5	9	9	9	7	7
15	0°	180°	7	7	7	5	5	9	9	9	7	7
16	0°	270°	7	7	7	5	5	9	9	9	7	7
17	90°	0°	7	7	7	5	5	1	1	1	1	1
18	90°	90°	7	7	7	5	5	1	1	1	1	1
19	90°	180°	7	7	7	5	5	1	1	1	1	1
20	90°	270°	7	7	7	5	5	1	1	1	1	1
21	90°	0°	7	7	7	5	5	3	3	3	3	3
22	90°	90°	7	7	7	5	5	3	3	3	3	3
23	90°	180°	7	7	7	5	5	3	3	3	3	3
24	90°	270°	7	7	7	5	5	3	3	3	3	3
25	90°	0°	7	7	7	5	5	5	5	5	7	7
26	90°	90°	7	7	7	5	5	5	5	5	7	7
27	90°	180°	7	7	7	5	5	5	5	5	7	7
28	90°	270°	7	7	7	5	5	5	5	5	7	7
29	90°	0°	7	7	7	5	5	9	9	9	7	7
30	90°	90°	7	7	7	5	5	9	9	9	7	7
31	90°	180°	7	7	7	5	5	9	9	9	7	7
32	90°	270°	7	7	7	5	5	9	9	9	7	7

Table C.1: Doubles Xconfig. stimuli numbering scheme part 1.

Doubles Xconfig. Stimuli Numbering Scheme Part 2												
			HotspotSeg					NonHotspotSeg				
Index	HotspotDir	NonHotspotDir	712R02	819R09	819R10	819R32	819R64	712R02	819R09	819R10	819R32	819R64
33	180°	0°	7	7	7	5	5	1	1	1	1	1
34	180°	90°	7	7	7	5	5	1	1	1	1	1
35	180°	180°	7	7	7	5	5	1	1	1	1	1
36	180°	270°	7	7	7	5	5	1	1	1	1	1
37	180°	0°	7	7	7	5	5	3	3	3	3	3
38	180°	90°	7	7	7	5	5	3	3	3	3	3
39	180°	180°	7	7	7	5	5	3	3	3	3	3
40	180°	270°	7	7	7	5	5	3	3	3	3	3
41	180°	0°	7	7	7	5	5	5	5	5	7	7
42	180°	90°	7	7	7	5	5	5	5	5	7	7
43	180°	180°	7	7	7	5	5	5	5	5	7	7
44	180°	270°	7	7	7	5	5	5	5	5	7	7
45	180°	0°	7	7	7	5	5	9	9	9	7	7
46	180°	90°	7	7	7	5	5	9	9	9	7	7
47	180°	180°	7	7	7	5	5	9	9	9	7	7
48	180°	270°	7	7	7	5	5	9	9	9	7	7
49	270°	0°	7	7	7	5	5	1	1	1	1	1
50	270°	90°	7	7	7	5	5	1	1	1	1	1
51	270°	180°	7	7	7	5	5	1	1	1	1	1
52	270°	270°	7	7	7	5	5	1	1	1	1	1
53	270°	0°	7	7	7	5	5	3	3	3	3	3
54	270°	90°	7	7	7	5	5	3	3	3	3	3
55	270°	180°	7	7	7	5	5	3	3	3	3	3
56	270°	270°	7	7	7	5	5	3	3	3	3	3
57	270°	0°	7	7	7	5	5	5	5	5	7	7
58	270°	90°	7	7	7	5	5	5	5	5	7	7
59	270°	180°	7	7	7	5	5	5	5	5	7	7
60	270°	270°	7	7	7	5	5	5	5	5	7	7
61	270°	0°	7	7	7	5	5	9	9	9	7	7
62	270°	90°	7	7	7	5	5	9	9	9	7	7
63	270°	180°	7	7	7	5	5	9	9	9	7	7
64	270°	270°	7	7	7	5	5	9	9	9	7	7

Table C.2: Doubles Xconfig. stimuli numbering scheme part 2.

Doubles Oconfig. Stimuli Numbering Scheme												
			HotspotSeg					NonHotspotSeg				
Index	HotspotDir	NonHotspotDir	712R02	819R09	819R10	819R32	819R64	712R02	819R09	819R10	819R32	819R64
65	0°	0°	8	8	8	8	2	2	2	2	2	4
66	0°	90°	8	8	8	8	2	2	2	2	2	4
67	0°	180°	8	8	8	8	2	2	2	2	2	4
68	0°	270°	8	8	8	8	2	2	2	2	2	4
69	0°	0°	8	8	8	8	2	4	4	4	4	4
70	0°	90°	8	8	8	8	2	4	4	4	4	4
71	0°	180°	8	8	8	8	2	4	4	4	4	4
72	0°	270°	8	8	8	8	2	4	4	4	4	4
73	0°	0°	8	8	8	8	2	6	6	6	6	6
74	0°	90°	8	8	8	8	2	6	6	6	6	6
75	0°	180°	8	8	8	8	2	6	6	6	6	6
76	0°	270°	8	8	8	8	2	6	6	6	6	6
77	90°	0°	8	8	8	8	2	2	2	2	2	4
78	90°	90°	8	8	8	8	2	2	2	2	2	4
79	90°	180°	8	8	8	8	2	2	2	2	2	4
80	90°	270°	8	8	8	8	2	2	2	2	2	4
81	90°	0°	8	8	8	8	2	4	4	4	4	4
82	90°	90°	8	8	8	8	2	4	4	4	4	4
83	90°	180°	8	8	8	8	2	4	4	4	4	4
84	90°	270°	8	8	8	8	2	4	4	4	4	4
85	90°	0°	8	8	8	8	2	6	6	6	6	6
86	90°	90°	8	8	8	8	2	6	6	6	6	6
87	90°	180°	8	8	8	8	2	6	6	6	6	6
88	90°	270°	8	8	8	8	2	6	6	6	6	6
89	180°	0°	8	8	8	8	2	2	2	2	2	4
90	180°	90°	8	8	8	8	2	2	2	2	2	4
91	180°	180°	8	8	8	8	2	2	2	2	2	4
92	180°	270°	8	8	8	8	2	2	2	2	2	4
93	180°	0°	8	8	8	8	2	4	4	4	4	4
94	180°	90°	8	8	8	8	2	4	4	4	4	4
95	180°	180°	8	8	8	8	2	4	4	4	4	4
96	180°	270°	8	8	8	8	2	4	4	4	4	4
97	180°	0°	8	8	8	8	2	6	6	6	6	6
98	180°	90°	8	8	8	8	2	6	6	6	6	6
99	180°	180°	8	8	8	8	2	6	6	6	6	6
100	180°	270°	8	8	8	8	2	6	6	6	6	6
101	270°	0°	8	8	8	8	2	2	2	2	2	4
102	270°	90°	8	8	8	8	2	2	2	2	2	4
103	270°	180°	8	8	8	8	2	2	2	2	2	4
104	270°	270°	8	8	8	8	2	2	2	2	2	4
105	270°	0°	8	8	8	8	2	4	4	4	4	4
106	270°	90°	8	8	8	8	2	4	4	4	4	4
107	270°	180°	8	8	8	8	2	4	4	4	4	4
108	270°	270°	8	8	8	8	2	4	4	4	4	4
109	270°	0°	8	8	8	8	2	6	6	6	6	6
110	270°	90°	8	8	8	8	2	6	6	6	6	6
111	270°	180°	8	8	8	8	2	6	6	6	6	6
112	270°	270°	8	8	8	8	2	6	6	6	6	6

Table C.3: Doubles Oconfig. stimuli numbering scheme.

## Appendix D

### Additional Results for Competitive Algorithm Applied to Singles Data

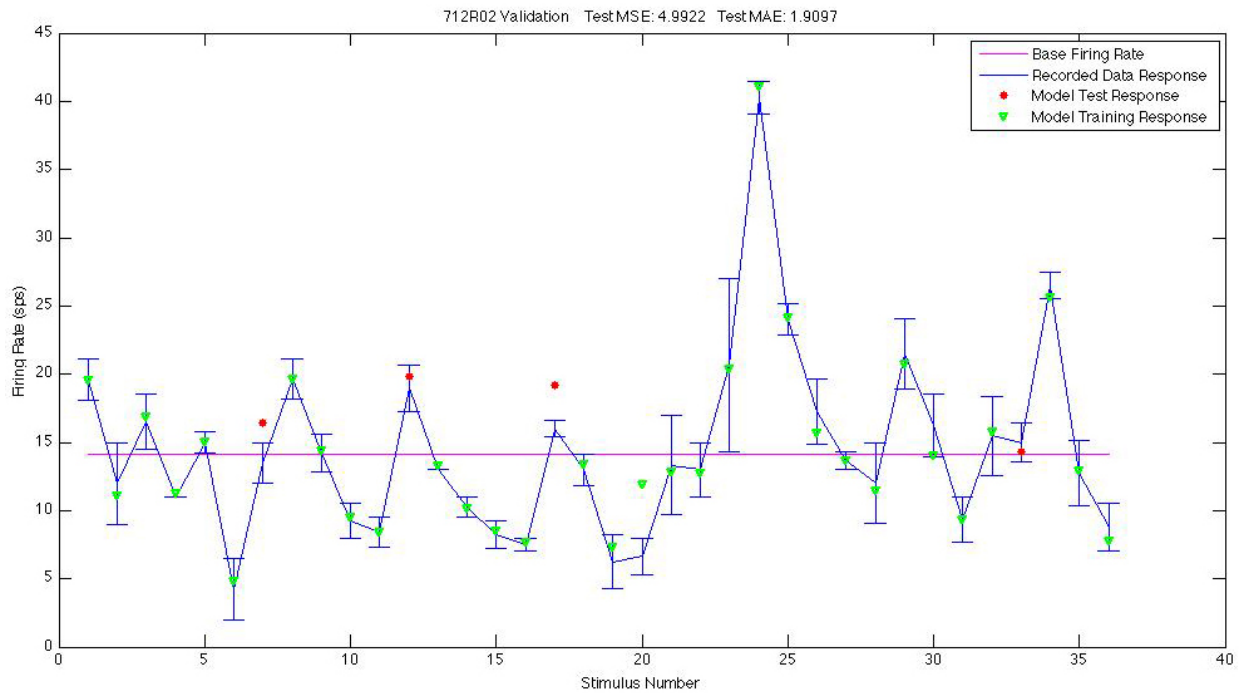


Figure D.1: Neuron 712R02 CA singles prediction.

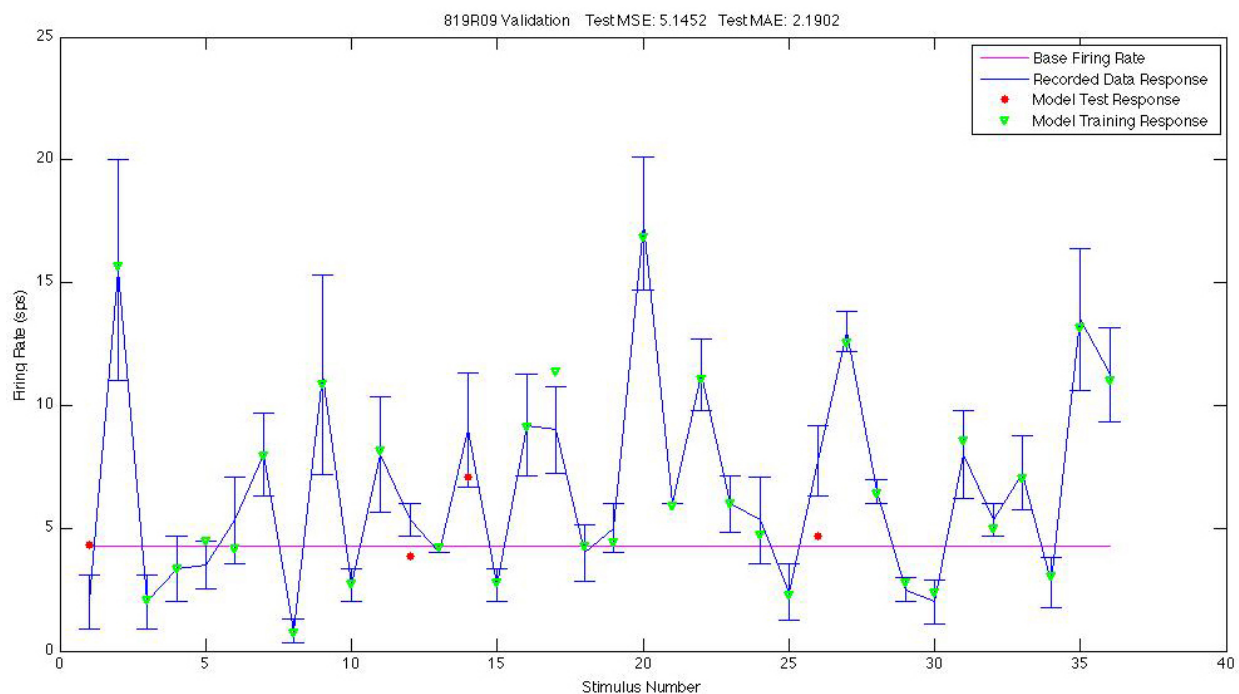


Figure D.2: Neuron 819R09 CA singles prediction.

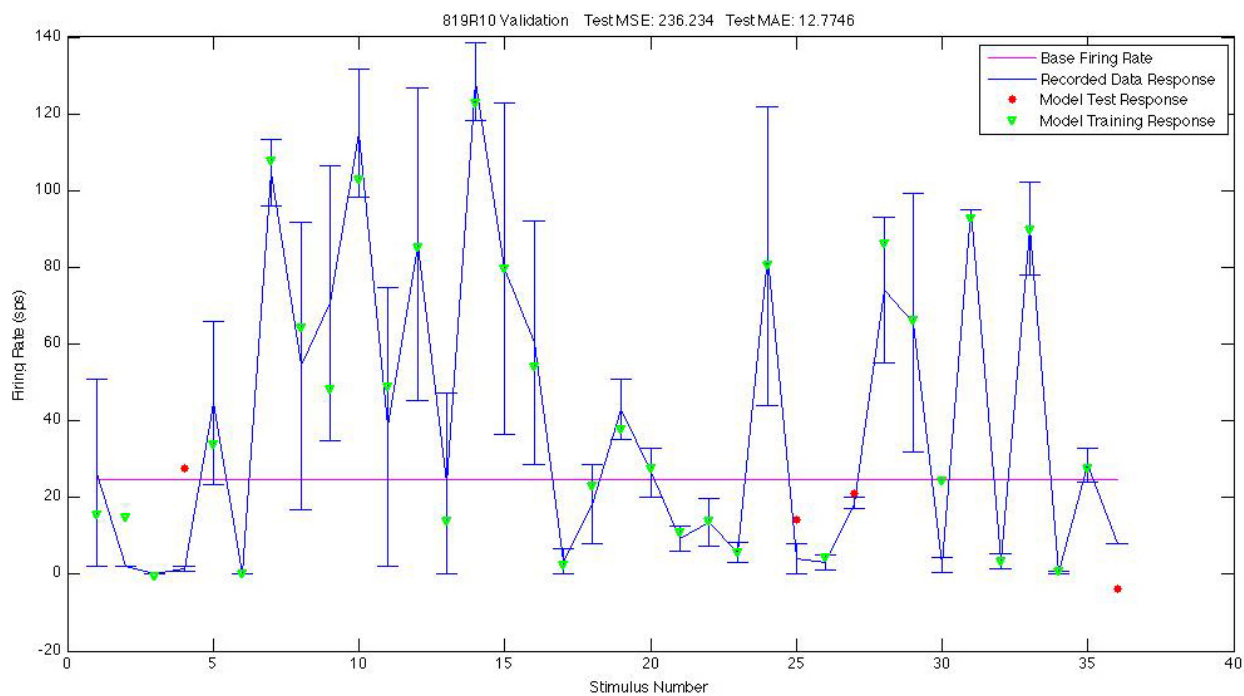


Figure D.3: Neuron 819R10 CA singles prediction.

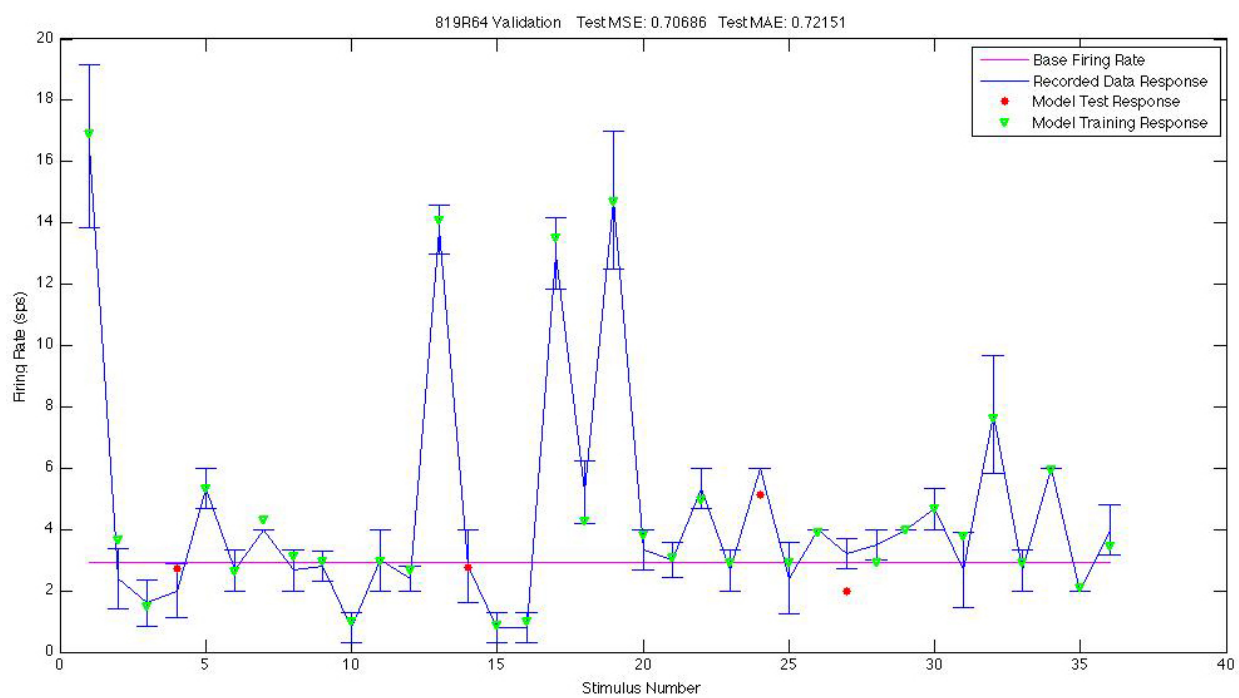


Figure D.4: Neuron 819R64 CA singles prediction.

## Appendix E

### Additional Results for Competitive Algorithm Applied to Doubles Data

Neuron 712R02 Doubles Subspace Results, PDF with Directional Covariation						
	Train			Test		
	MSE	MAE	MRE	MSE	MAE	MRE
Xconfig. 0°	2.8815	1.0812	0.0000	9.8349	3.1262	0.3780
Xconfig. 90°	2.7455	1.1500	0.2511	11.0317	2.6523	0.7359
Xconfig. 180°	0.3455	0.4103	0.0000	1.2862	1.1213	0.2372
Xconfig. 270°	0.5405	0.5966	0.0044	1.3185	0.8242	0.0653
Oconfig. 0°	0.5652	0.6079	0.0205	1.4780	1.1871	0.0996
Oconfig. 90°	2.5867	0.9949	0.1609	11.4678	2.4628	0.1436
Oconfig. 180°	0.3757	0.4186	0.0000	0.0635	0.2516	0.1321
Oconfig. 270°	0.2891	0.3028	0.0286	1.5467	1.0313	0.0372
	MaxFiringRate		NumGauss	NumGenetic	NumGreedy	
Xconfig. 0°	21.75		4	0	4	
Xconfig. 90°	41.25		4	1	3	
Xconfig. 180°	31.00		3	2	1	
Xconfig. 270°	32.00		4	4	0	
Oconfig. 0°	18.33		3	2	1	
Oconfig. 90°	41.00		4	2	2	
Oconfig. 180°	28.25		3	1	2	
Oconfig. 270°	19.00		4	3	1	

Table E.1: Neuron 712R02 CA doubles subspace prediction results.

Neuron 819R09 Doubles Subspace Results, PDF with Directional Covariation						
	Train			Test		
	MSE	MAE	MRE	MSE	MAE	MRE
Xconfig. 0°	2.1841	1.1348	0.3810	4.5221	2.0585	0.0543
Xconfig. 90°	0.3960	0.4204	0.0794	1.1625	0.8856	0.3503
Xconfig. 180°	0.0351	0.1493	0.0398	0.4532	0.6232	0.2899
Xconfig. 270°	2.4508	1.1012	0.3289	1.5442	1.1580	0.0844
Oconfig. 0°	0.7372	0.3594	0.0579	0.2785	0.4045	0.0830
Oconfig. 90°	0.3824	0.4624	0.0278	7.9387	2.8101	1.4921
Oconfig. 180°	0.9338	0.7042	0.0000	0.5530	0.7435	0.0000
Oconfig. 270°	0.7850	0.5978	0.0092	10.1481	2.5884	0.0000
	MaxFiringRate		NumGauss	NumGenetic		NumGreedy
Xconfig. 0°	21.33		4	4		0
Xconfig. 90°	8.20		4	3		1
Xconfig. 180°	5.00		4	2		2
Xconfig. 270°	17.33		4	1		3
Oconfig. 0°	9.50		4	2		2
Oconfig. 90°	25.00		3	1		2
Oconfig. 180°	18.33		3	1		2
Oconfig. 270°	44.67		4	2		2

Table E.2: Neuron 819R09 CA doubles subspace prediction results.

Neuron 819R10 Doubles Subspace Results, PDF with Directional Covariation						
	Train			Test		
	MSE	MAE	MRE	MSE	MAE	MRE
Xconfig. 0°	133.3170	7.9590	3.6848	57.6034	7.5830	2.8058
Xconfig. 90°	34.6145	3.4000	0.2399	35.8151	4.2674	2.2733
Xconfig. 180°	44.1839	4.0961	0.5009	64.0221	7.8158	0.0000
Xconfig. 270°	173.2875	8.4744	2.2590	143.4726	11.6698	6.3910
Oconfig. 0°	300.6654	9.5033	0.2934	269.6443	15.7341	9.5674
Oconfig. 90°	7.5006	1.8001	0.0000	29.3500	3.9710	0.0000
Oconfig. 180°	180.0253	9.3605	0.1920	289.9442	15.4254	2.2738
Oconfig. 270°	620.2273	16.7984	0.0000	23.1743	3.7790	0.0000
	MaxFiringRate		NumGauss	NumGenetic		NumGreedy
Xconfig. 0°	164.00		4	1		3
Xconfig. 90°	72.25		4	1		3
Xconfig. 180°	70.00		4	2		2
Xconfig. 270°	127.67		4	1		3
Oconfig. 0°	90.67		4	1		3
Oconfig. 90°	114.33		4	2		2
Oconfig. 180°	121.67		4	2		2
Oconfig. 270°	127.67		3	0		3

Table E.3: Neuron 819R10 CA doubles subspace prediction results.



Neuron 819R32 Doubles Subspace Results, PDF with Directional Covariation						
	Train			Test		
	MSE	MAE	MRE	MSE	MAE	MRE
Xconfig. 0°	0.3785	0.4965	0.1086	0.0265	0.1559	0.0000
Xconfig. 90°	1.0562	0.7055	0.1141	1.6359	0.9705	0.6131
Xconfig. 180°	0.9896	0.7024	0.0683	0.4984	0.6682	0.4481
Xconfig. 270°	0.4142	0.4763	0.0000	0.3513	0.4532	0.0000
Oconfig. 0°	0.3459	0.3380	0.0882	0.1035	0.2804	0.0613
Oconfig. 90°	2.0427	0.6801	0.4848	5.3253	2.0872	0.9450
Oconfig. 180°	0.1438	0.2646	0.0289	1.0923	1.0030	0.0000
Oconfig. 270°	0.0774	0.2196	0.0072	0.4776	0.5258	0.0000
	MaxFiringRate		NumGauss	NumGenetic	NumGreedy	
Xconfig. 0°	15.33		4	2	2	
Xconfig. 90°	33.80		4	3	1	
Xconfig. 180°	24.00		4	2	2	
Xconfig. 270°	17.75		3	1	2	
Oconfig. 0°	6.75		4	2	2	
Oconfig. 90°	22.00		4	2	2	
Oconfig. 180°	12.80		4	2	2	
Oconfig. 270°	6.00		4	1	3	

Table E.4: Neuron 819R32 CA doubles subspace prediction results.

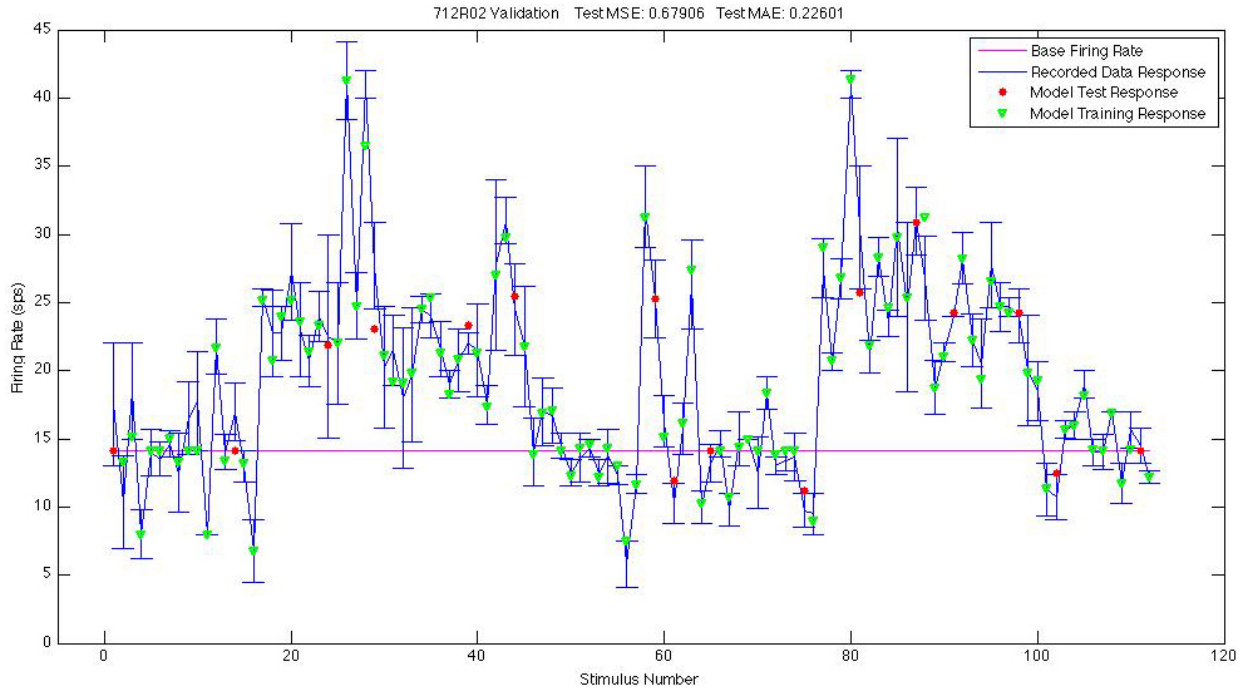


Figure E.1: Neuron 712R02 CA doubles prediction. Combination of subspace results.

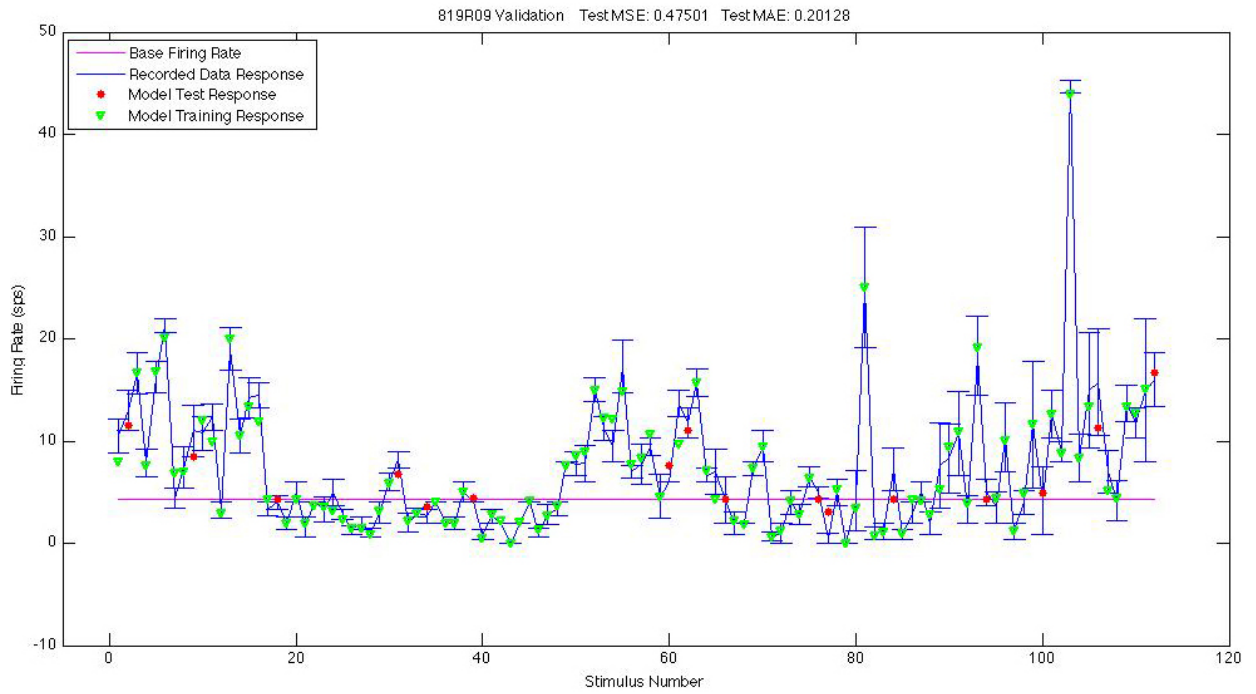


Figure E.2: Neuron 819R09 CA doubles prediction. Combination of subspace results.

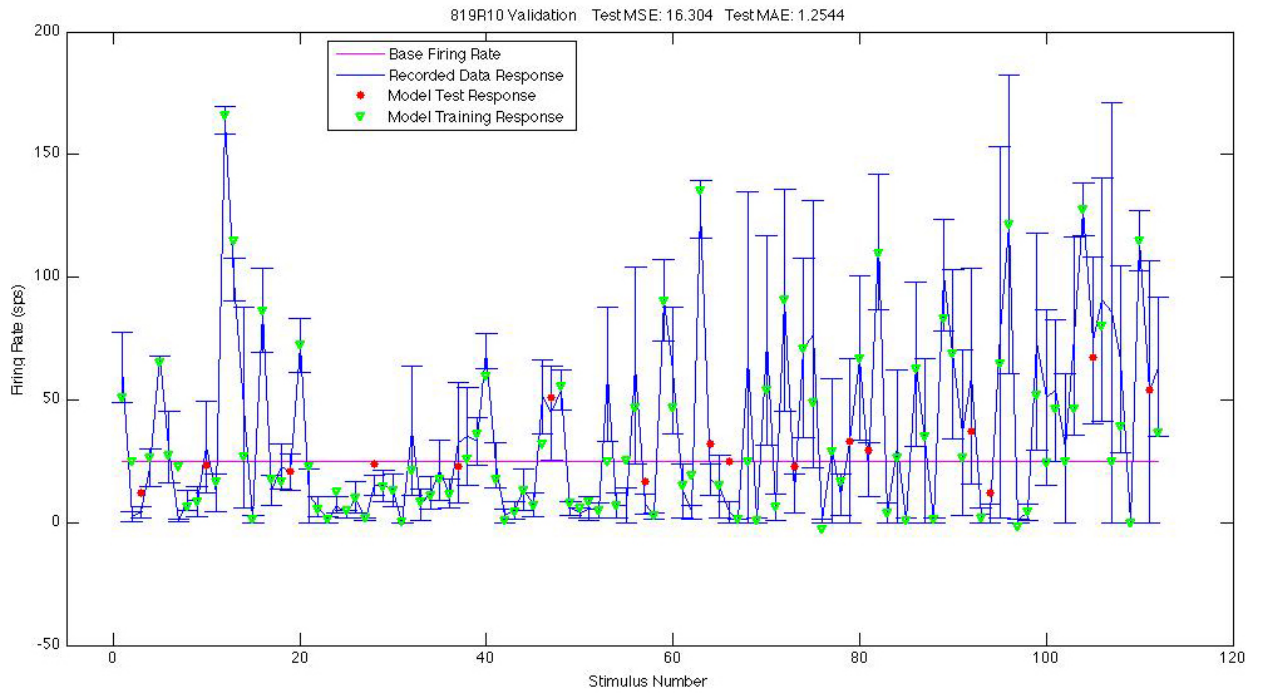


Figure E.3: Neuron 819R10 CA doubles prediction. Combination of subspace results.

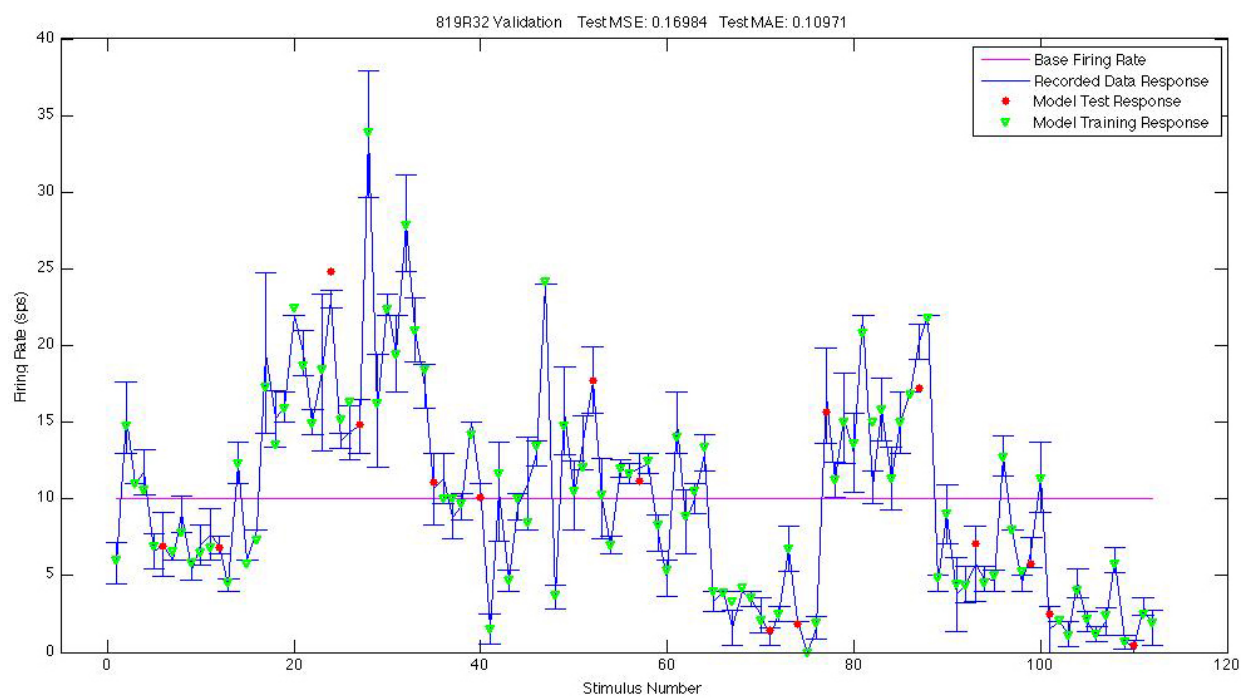


Figure E.4: Neuron 819R32 CA doubles prediction. Combination of subspace results.

## Appendix F

### Additional Results for Linear Regression Technique Applied to Flow Data

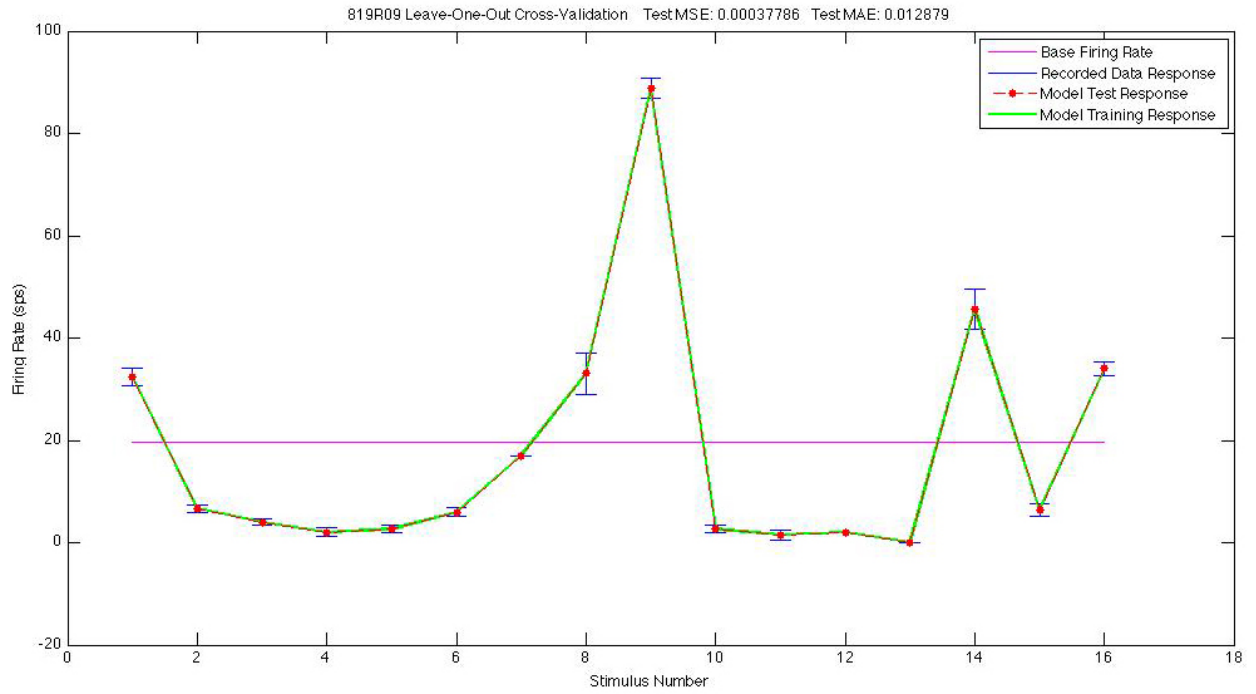


Figure F.1: Neuron 819R09 LR flow prediction using best T3-FS4 combination.

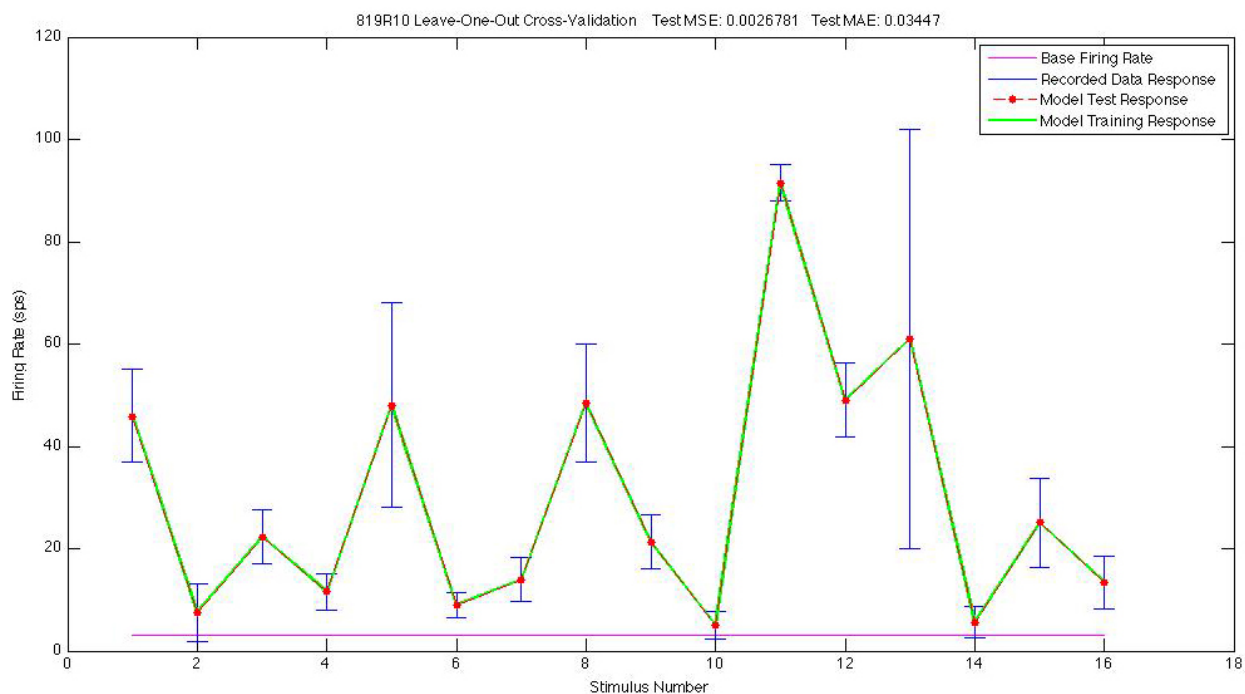


Figure F.2: Neuron 819R10 LR flow prediction using best T3-FS4 combination.

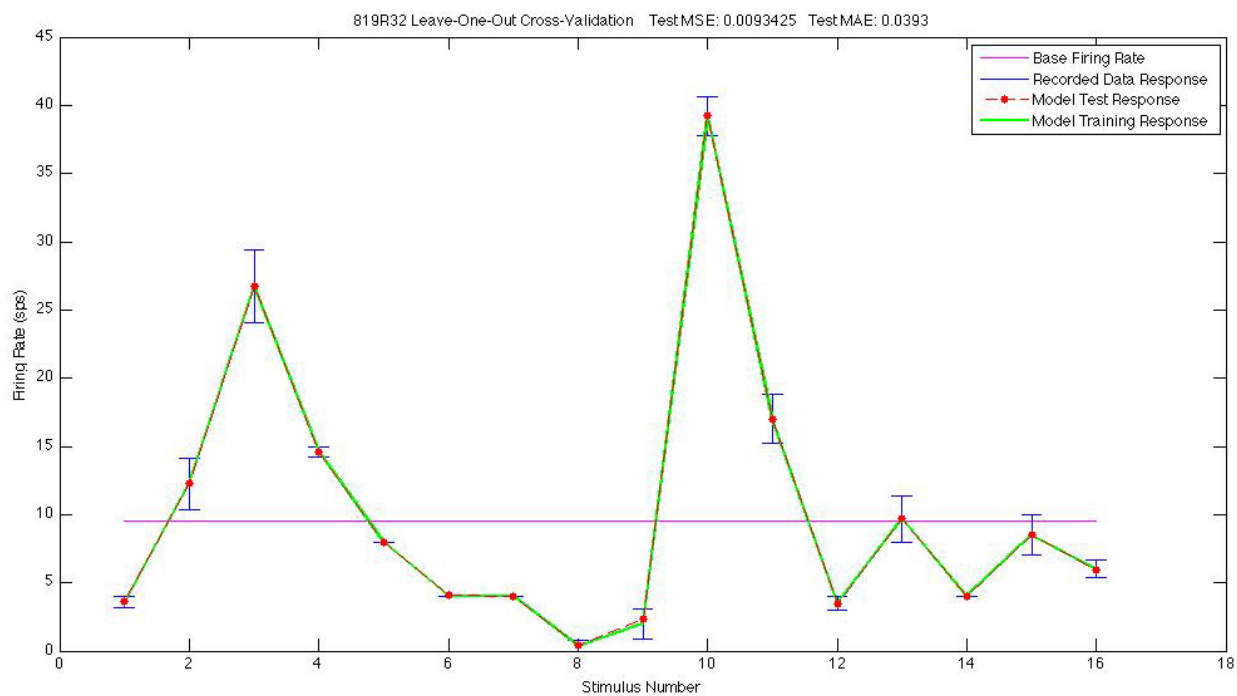


Figure F.3: Neuron 819R32 LR flow prediction using best T3-FS4 combination.

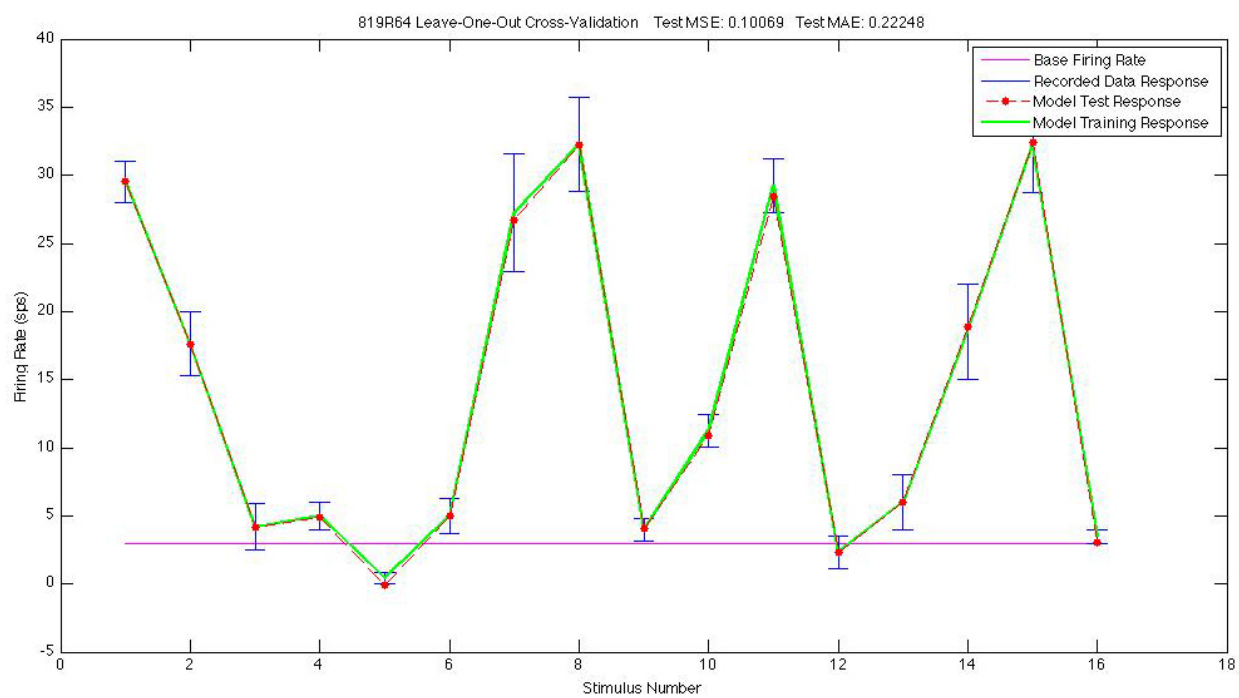


Figure F.4: Neuron 819R64 LR flow prediction using best T3-FS4 combination.

## Appendix G

### Additional Flow Data Linear Model Feature Weight Analysis

T3-FS4 Feature Indices	
1	Xconfig. hotspot contrib.
2	Xconfig. sin hotspot dir. * hotspot contrib.
3	Xconfig. cos hotspot dir. * hotspot contrib.
4	Oconfig. hotspot contrib.
5	Oconfig. sin hotspot dir. * hotspot contrib.
6	Oconfig. cos hotspot dir. * hotspot contrib.
7	sin segment 1 dir. * segment 1 contrib.
8	cos segment 1 dir. * segment 1 contrib.
9	sin segment 2 dir. * segment 2 contrib.
10	cos segment 2 dir. * segment 2 contrib.
11	sin segment 3 dir. * segment 3 contrib.
12	cos segment 3 dir. * segment 3 contrib.
13	sin segment 4 dir. * segment 4 contrib.
14	cos segment 4 dir. * segment 4 contrib.
15	sin segment 5 dir. * segment 5 contrib.
16	cos segment 5 dir. * segment 5 contrib.
17	sin segment 6 dir. * segment 6 contrib.
18	cos segment 6 dir. * segment 6 contrib.
19	sin segment 7 dir. * segment 7 contrib.
20	cos segment 7 dir. * segment 7 contrib.
21	sin segment 8 dir. * segment 8 contrib.
22	cos segment 8 dir. * segment 8 contrib.
23	sin segment 9 dir. * segment 9 contrib.
24	cos segment 9 dir. * segment 9 contrib.

Table G.1: T3-FS4 feature indices.



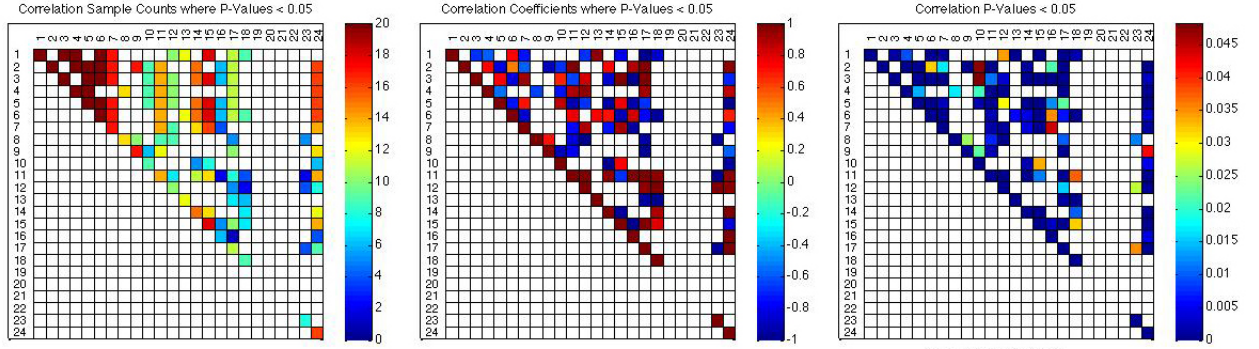


Figure G.1: Neuron 819R09 LR flow prediction feature weight correlations for best T3-FS4 combinations.

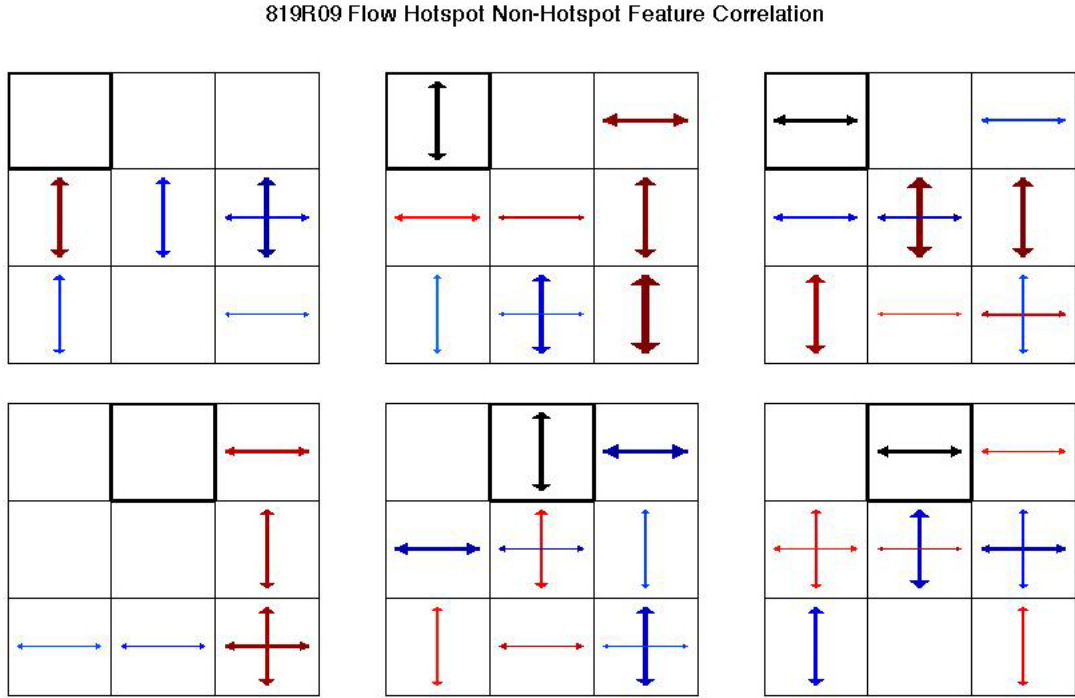


Figure G.2: Neuron 819R09 LR flow prediction feature weight correlation visualization for best T3-FS4 combinations.



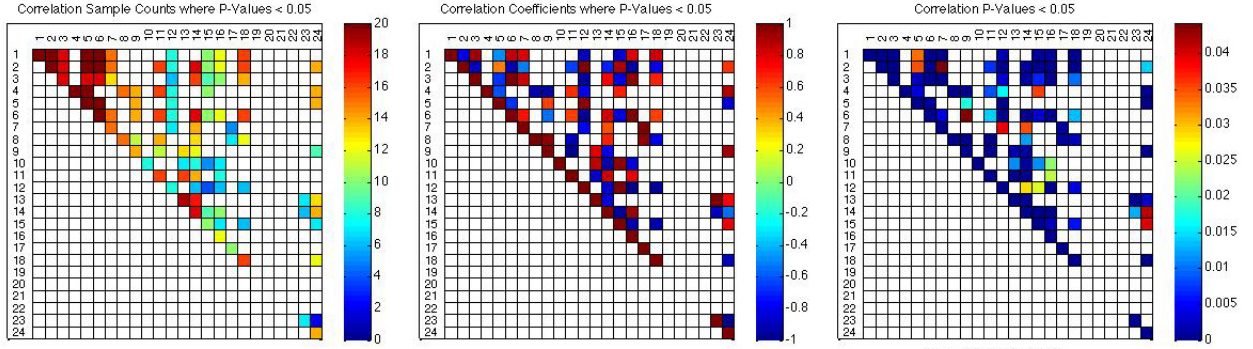


Figure G.3: Neuron 819R10 LR flow prediction feature weight correlations for best T3-FS4 combinations.

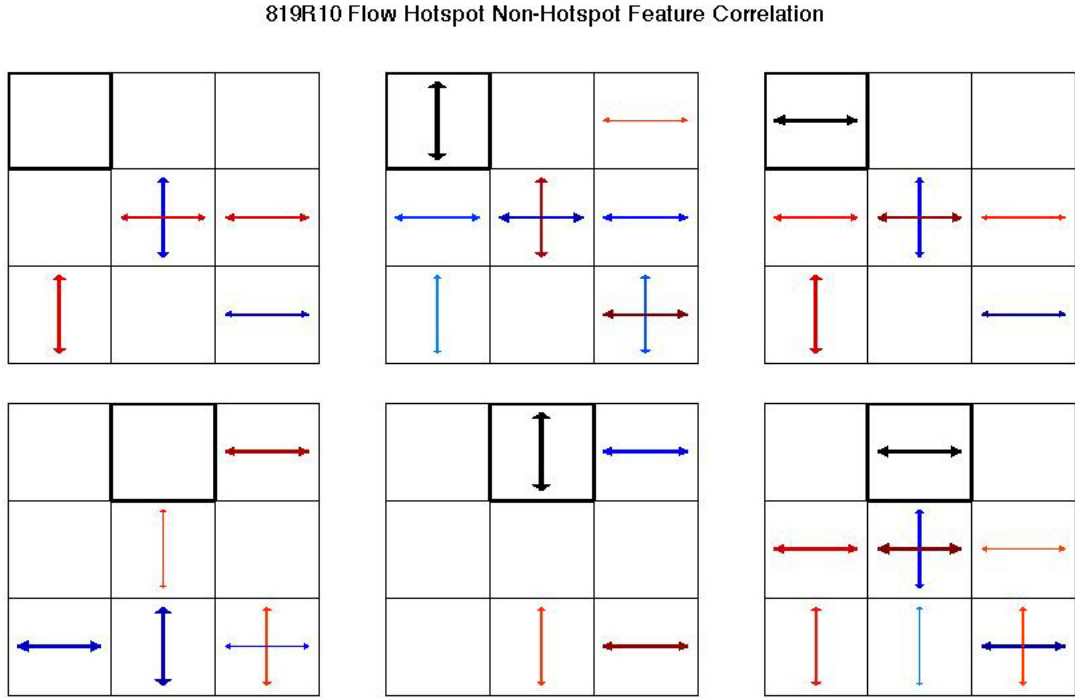


Figure G.4: Neuron 819R10 LR flow prediction feature weight correlation visualization for best T3-FS4 combinations.

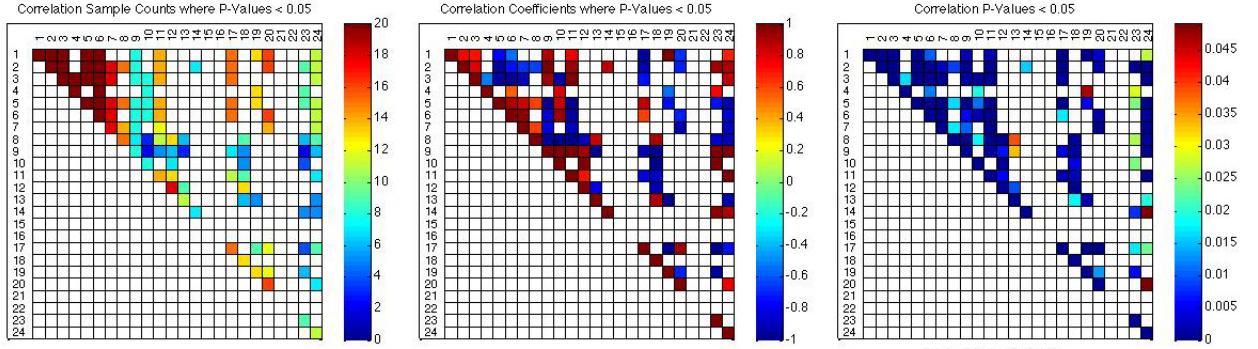


Figure G.5: Neuron 819R32 LR flow prediction feature weight correlations for best T3-FS4 combinations.

819R32 Flow Hotspot Non-Hotspot Feature Correlation

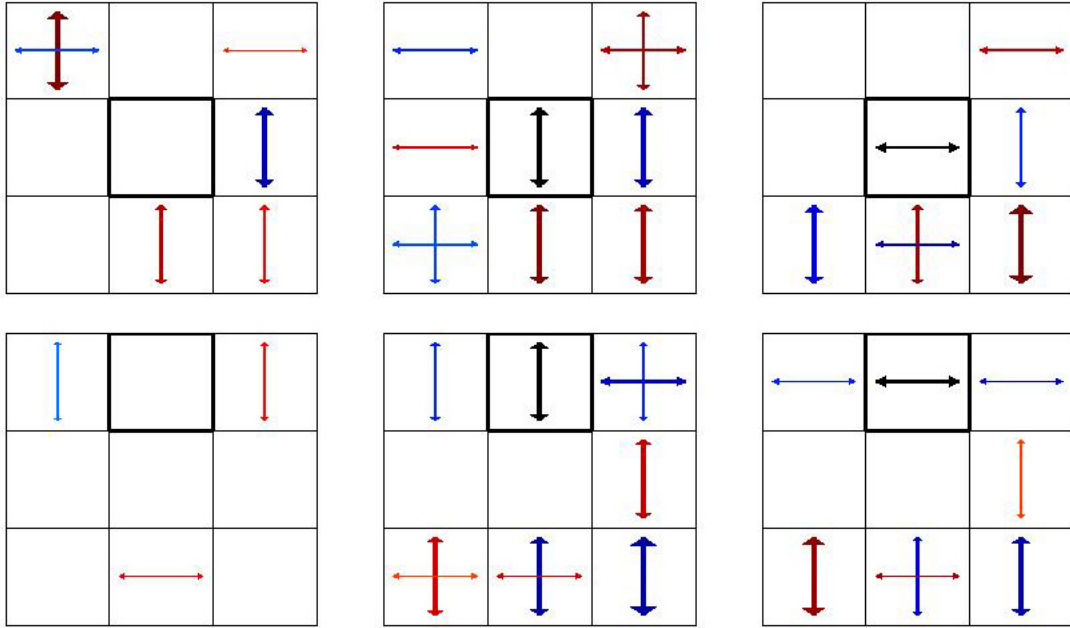


Figure G.6: Neuron 819R32 LR flow prediction feature weight correlation visualization for best T3-FS4 combinations.

## Appendix H

### Feature Set Combinations Used for Flow Data Linear Model Feature Weight Analysis

Neuron 712R02 Flow Results, Best Combinations of T3-FS4 Features Used for Linear Model Feature Weight Analysis																									
Index	TestMSE	Xconfg. hotspot contrib.	Xconfg. sin hotspot dir. * hotspot contrib.	Xconfg. cos hotspot dir. * hotspot contrib.	Oconfg. hotspot contrib.	Oconfg. sin hotspot dir. * hotspot contrib.	Oconfg. cos hotspot dir. * hotspot contrib.	sin seg. 1 dir. * seg. 1 contrib.	cos seg. 1 dir. * seg. 1 contrib.	sin seg. 2 dir. * seg. 2 contrib.	cos seg. 2 dir. * seg. 2 contrib.	sin seg. 3 dir. * seg. 3 contrib.	cos seg. 3 dir. * seg. 3 contrib.	sin seg. 4 dir. * seg. 4 contrib.	cos seg. 4 dir. * seg. 4 contrib.	sin seg. 5 dir. * seg. 5 contrib.	cos seg. 5 dir. * seg. 5 contrib.	sin seg. 6 dir. * seg. 6 contrib.	cos seg. 6 dir. * seg. 6 contrib.	sin seg. 7 dir. * seg. 7 contrib.	cos seg. 7 dir. * seg. 7 contrib.	sin seg. 8 dir. * seg. 8 contrib.	cos seg. 8 dir. * seg. 8 contrib.	sin seg. 9 dir. * seg. 9 contrib.	cos seg. 9 dir. * seg. 9 contrib.
		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1	0.0393	X	X	X	X	X	X	-	X	X	X	-	X	X	X	X	-	X	X	X	+	+	+	+	X
2	0.1268	X	X	X	X	X	X	-	X	X	X	-	X	X	X	X	-	X	X	X	+	+	+	+	X
3	1.4378	X	X	X	X	X	X	X	X	X	X	-	X	X	X	X	-	X	X	-	+	+	+	+	X
4	2.1130	X	X	X	X	X	X	X	X	-	X	-	X	X	X	X	X	X	-	-	+	+	+	+	X
5	2.2023	X	X	X	X	X	X	X	X	X	X	-	X	X	X	X	X	X	-	-	+	+	+	+	X
6	6.1371	X	X	X	X	X	X	-	X	X	X	-	X	X	X	X	X	X	X	X	+	+	+	+	X
7	9.3007	X	X	X	X	X	X	X	X	-	X	-	X	X	X	X	-	X	X	X	+	+	+	+	X
8	9.4612	X	X	X	X	X	X	-	X	X	X	-	X	X	X	X	-	X	X	-	+	+	+	+	X
9	10.2988	X	X	X	X	X	X	X	X	X	X	-	X	X	X	X	-	X	X	X	+	+	+	+	X
10	10.4693	X	X	X	X	X	X	X	X	X	X	-	X	X	X	X	-	X	X	X	+	+	+	+	X
11	11.2766	X	X	X	X	X	X	X	X	X	X	-	X	X	X	X	-	X	X	-	+	+	+	+	X
12	11.3529	X	X	X	X	X	X	X	X	X	X	-	X	X	X	X	-	X	X	-	+	+	+	+	X
13	12.5818	X	X	X	X	X	X	X	X	X	X	-	X	X	X	X	-	X	X	X	+	+	+	+	X
14	15.6500	X	X	X	X	X	X	X	X	X	X	-	X	X	X	X	-	X	X	X	+	+	+	+	X
15	17.0402	X	X	X	X	X	X	X	X	X	X	-	X	X	X	X	-	X	X	-	+	+	+	+	X
16	17.0989	X	X	X	X	X	X	X	X	X	X	-	X	X	X	X	-	X	X	-	+	+	+	+	X
17	17.5022	X	X	X	X	X	X	X	X	X	X	-	X	X	X	X	-	X	X	-	+	+	+	+	X
18	17.8549	X	X	X	X	X	X	X	X	X	X	-	X	X	X	X	-	X	X	-	+	+	+	+	X
19	18.5284	X	X	X	X	X	X	X	X	X	X	-	X	X	X	X	-	X	X	-	+	+	+	+	X
20	19.0785	X	X	X	X	X	X	X	X	X	X	-	X	X	X	X	-	X	X	-	+	+	+	+	X

Table H.1: Neuron 712R02 best combinations of T3-FS4 features used for flow data linear model feature weight analysis. - denotes features not selected. x denotes features selected. + denotes features associated with hotspot locations and thus were not available for non-hotspot feature selection.

Neuron 819R09 Flow Results, Best Combinations of T3-FS4 Features Used for Linear Model Feature Weight Analysis																										
Index	TestMSE	Xconfg. hotspot contrib.	Xconfg. sin hotspot dir. * hotspot contrib.	Xconfg. cos hotspot dir. * hotspot contrib.	Ocnfg. hotspot contrib.	Ocnfg. sin hotspot dir. * hotspot contrib.	Ocnfg. cos hotspot dir. * hotspot contrib.	sin seg. 1 dir. * seg. 1 contrib.	cos seg. 1 dir. * seg. 1 contrib.	sin seg. 2 dir. * seg. 2 contrib.	cos seg. 2 dir. * seg. 2 contrib.	sin seg. 3 dir. * seg. 3 contrib.	cos seg. 3 dir. * seg. 3 contrib.	sin seg. 4 dir. * seg. 4 contrib.	cos seg. 4 dir. * seg. 4 contrib.	sin seg. 5 dir. * seg. 5 contrib.	cos seg. 5 dir. * seg. 5 contrib.	sin seg. 6 dir. * seg. 6 contrib.	cos seg. 6 dir. * seg. 6 contrib.	sin seg. 7 dir. * seg. 7 contrib.	cos seg. 7 dir. * seg. 7 contrib.	sin seg. 8 dir. * seg. 8 contrib.	cos seg. 8 dir. * seg. 8 contrib.	sin seg. 9 dir. * seg. 9 contrib.	cos seg. 9 dir. * seg. 9 contrib.	
		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	+	+	+	+	+	+
1	0.0004	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	-	-	x	-	+ <td>+<td>+<td>+<td>x</td><td>-</td></td></td></td>	+ <td>+<td>+<td>x</td><td>-</td></td></td>	+ <td>+<td>x</td><td>-</td></td>	+ <td>x</td> <td>-</td>	x	-
2	0.1196	x	x	x	x	x	x	-	x	x	x	-	x	x	x	x	-	-	x	-	+ <td>+<td>+<td>+<td>x</td><td>-</td></td></td></td>	+ <td>+<td>+<td>x</td><td>-</td></td></td>	+ <td>+<td>x</td><td>-</td></td>	+ <td>x</td> <td>-</td>	x	-
3	0.5258	x	x	x	x	x	x	x	x	x	x	-	x	x	x	x	-	-	x	-	+ <td>+<td>+<td>+<td>x</td><td>-</td></td></td></td>	+ <td>+<td>+<td>x</td><td>-</td></td></td>	+ <td>+<td>x</td><td>-</td></td>	+ <td>x</td> <td>-</td>	x	-
4	1.0454	x	x	x	x	x	x	x	x	x	x	-	x	x	x	x	-	-	x	-	+ <td>+<td>+<td>+<td>x</td><td>-</td></td></td></td>	+ <td>+<td>+<td>x</td><td>-</td></td></td>	+ <td>+<td>x</td><td>-</td></td>	+ <td>x</td> <td>-</td>	x	-
5	1.3143	x	x	x	x	x	x	x	x	x	x	-	x	x	x	x	-	-	x	-	+ <td>+<td>+<td>+<td>x</td><td>-</td></td></td></td>	+ <td>+<td>+<td>x</td><td>-</td></td></td>	+ <td>+<td>x</td><td>-</td></td>	+ <td>x</td> <td>-</td>	x	-
6	1.6515	x	x	x	x	x	x	x	x	x	x	-	x	x	x	x	-	-	x	-	+ <td>+<td>+<td>+<td>x</td><td>-</td></td></td></td>	+ <td>+<td>+<td>x</td><td>-</td></td></td>	+ <td>+<td>x</td><td>-</td></td>	+ <td>x</td> <td>-</td>	x	-
7	1.8586	x	x	x	x	x	x	x	x	x	x	-	x	x	x	x	-	-	x	-	+ <td>+<td>+<td>+<td>x</td><td>-</td></td></td></td>	+ <td>+<td>+<td>x</td><td>-</td></td></td>	+ <td>+<td>x</td><td>-</td></td>	+ <td>x</td> <td>-</td>	x	-
8	2.0731	x	x	x	x	x	x	-	x	x	x	-	x	x	x	x	-	-	x	-	+ <td>+<td>+<td>+<td>x</td><td>-</td></td></td></td>	+ <td>+<td>+<td>x</td><td>-</td></td></td>	+ <td>+<td>x</td><td>-</td></td>	+ <td>x</td> <td>-</td>	x	-
9	2.2870	x	x	x	x	x	x	-	x	x	x	-	x	x	x	x	-	-	x	-	+ <td>+<td>+<td>+<td>x</td><td>-</td></td></td></td>	+ <td>+<td>+<td>x</td><td>-</td></td></td>	+ <td>+<td>x</td><td>-</td></td>	+ <td>x</td> <td>-</td>	x	-
10	2.5993	x	x	x	x	x	x	x	x	x	x	-	x	x	x	x	-	-	x	-	+ <td>+<td>+<td>+<td>x</td><td>-</td></td></td></td>	+ <td>+<td>+<td>x</td><td>-</td></td></td>	+ <td>+<td>x</td><td>-</td></td>	+ <td>x</td> <td>-</td>	x	-
11	4.0643	x	x	x	x	x	x	-	x	x	x	-	x	x	x	x	-	-	x	-	+ <td>+<td>+<td>+<td>x</td><td>-</td></td></td></td>	+ <td>+<td>+<td>x</td><td>-</td></td></td>	+ <td>+<td>x</td><td>-</td></td>	+ <td>x</td> <td>-</td>	x	-
12	4.1349	x	x	x	x	x	x	x	x	x	x	-	x	x	x	x	-	-	x	-	+ <td>+<td>+<td>+<td>x</td><td>-</td></td></td></td>	+ <td>+<td>+<td>x</td><td>-</td></td></td>	+ <td>+<td>x</td><td>-</td></td>	+ <td>x</td> <td>-</td>	x	-
13	5.7829	x	x	x	x	x	x	x	x	x	x	-	x	x	x	x	-	-	x	-	+ <td>+<td>+<td>+<td>x</td><td>-</td></td></td></td>	+ <td>+<td>+<td>x</td><td>-</td></td></td>	+ <td>+<td>x</td><td>-</td></td>	+ <td>x</td> <td>-</td>	x	-
14	6.4346	x	x	x	x	x	x	-	x	x	x	-	x	x	x	x	-	-	x	-	+ <td>+<td>+<td>+<td>x</td><td>-</td></td></td></td>	+ <td>+<td>+<td>x</td><td>-</td></td></td>	+ <td>+<td>x</td><td>-</td></td>	+ <td>x</td> <td>-</td>	x	-
15	6.5984	x	x	x	x	x	x	-	x	x	x	-	x	x	x	x	-	-	x	-	+ <td>+<td>+<td>+<td>x</td><td>-</td></td></td></td>	+ <td>+<td>+<td>x</td><td>-</td></td></td>	+ <td>+<td>x</td><td>-</td></td>	+ <td>x</td> <td>-</td>	x	-
16	8.1070	x	x	x	x	x	x	-	x	x	x	-	x	x	x	x	-	-	x	-	+ <td>+<td>+<td>+<td>x</td><td>-</td></td></td></td>	+ <td>+<td>+<td>x</td><td>-</td></td></td>	+ <td>+<td>x</td><td>-</td></td>	+ <td>x</td> <td>-</td>	x	-
17	8.4968	x	x	x	x	x	x	x	x	x	x	-	x	x	x	x	-	-	x	-	+ <td>+<td>+<td>+<td>x</td><td>-</td></td></td></td>	+ <td>+<td>+<td>x</td><td>-</td></td></td>	+ <td>+<td>x</td><td>-</td></td>	+ <td>x</td> <td>-</td>	x	-
18	9.6572	x	x	x	x	x	x	x	x	x	x	-	x	x	x	x	-	-	x	-	+ <td>+<td>+<td>+<td>x</td><td>-</td></td></td></td>	+ <td>+<td>+<td>x</td><td>-</td></td></td>	+ <td>+<td>x</td><td>-</td></td>	+ <td>x</td> <td>-</td>	x	-
19	9.7083	x	x	x	x	x	x	x	x	x	x	-	x	x	x	x	-	-	x	-	+ <td>+<td>+<td>+<td>x</td><td>-</td></td></td></td>	+ <td>+<td>+<td>x</td><td>-</td></td></td>	+ <td>+<td>x</td><td>-</td></td>	+ <td>x</td> <td>-</td>	x	-
20	10.3209	x	x	x	x	x	x	x	x	x	x	-	x	x	x	x	-	-	x	-	+ <td>+<td>+<td>+<td>x</td><td>-</td></td></td></td>	+ <td>+<td>+<td>x</td><td>-</td></td></td>	+ <td>+<td>x</td><td>-</td></td>	+ <td>x</td> <td>-</td>	x	-

Table H.2: Neuron 819R09 best combinations of T3-FS4 features used for flow data linear model feature weight analysis. - denotes features not selected. x denotes features selected. + denotes features associated with hotspot locations and thus were not available for non-hotspot feature selection.

Neuron 819R10 Flow Results, Best Combinations of T3-FS4 Features Used for Linear Model Feature Weight Analysis																									
Index	TestMSE	Xconfg. hotspot contrib.	Xconfg. sin hotspot dir. * hotspot contrib.	Xconfg. cos hotspot dir. * hotspot contrib.	Ocnfg. hotspot contrib.	Ocnfg. sin hotspot dir. * hotspot contrib.	Ocnfg. cos hotspot dir. * hotspot contrib.	sin seg. 1 dir. * seg. 1 contrib.	cos seg. 1 dir. * seg. 1 contrib.	sin seg. 2 dir. * seg. 2 contrib.	cos seg. 2 dir. * seg. 2 contrib.	sin seg. 3 dir. * seg. 3 contrib.	cos seg. 3 dir. * seg. 3 contrib.	sin seg. 4 dir. * seg. 4 contrib.	cos seg. 4 dir. * seg. 4 contrib.	sin seg. 5 dir. * seg. 5 contrib.	cos seg. 5 dir. * seg. 5 contrib.	sin seg. 6 dir. * seg. 6 contrib.	cos seg. 6 dir. * seg. 6 contrib.	sin seg. 7 dir. * seg. 7 contrib.	cos seg. 7 dir. * seg. 7 contrib.	sin seg. 8 dir. * seg. 8 contrib.	cos seg. 8 dir. * seg. 8 contrib.	sin seg. 9 dir. * seg. 9 contrib.	cos seg. 9 dir. * seg. 9 contrib.
		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
1	0.0027	x	x	x	x	x	x	x	x	x	-	x	-	x	x	x	-	x	-	x	+	+	+	+	x
2	0.0699	x	x	x	x	x	x	x	-	x	x	-	x	x	x	x	x	x	x	x	+	+	+	+	x
3	0.7876	x	x	x	x	x	x	x	-	x	x	-	x	x	x	x	-	x	-	x	+	+	+	+	x
4	1.3537	x	x	x	x	x	x	x	-	x	x	-	x	x	x	x	-	x	-	x	+	+	+	+	x
5	1.5793	x	x	x	x	x	x	x	x	x	-	-	x	x	x	x	-	x	-	x	+	+	+	+	x
6	2.3476	x	x	x	x	x	x	x	x	x	-	-	x	x	x	x	-	x	-	x	+	+	+	+	x
7	3.1541	x	x	x	x	x	x	x	x	x	-	-	x	x	x	x	-	x	-	x	+	+	+	+	x
8	3.4955	x	x	x	x	x	x	x	x	x	-	-	x	x	x	x	-	x	-	x	+	+	+	+	x
9	6.7259	x	x	x	x	x	x	x	x	x	-	-	x	x	x	x	-	x	-	x	+	+	+	+	x
10	7.5221	x	x	x	x	x	x	-	-	x	x	-	x	x	x	x	-	x	-	x	+	+	+	+	x
11	17.3350	x	x	x	x	x	x	x	x	-	-	x	x	x	x	x	-	x	-	x	+	+	+	+	x
12	24.4009	x	x	x	x	x	x	x	x	-	-	x	x	x	x	x	-	x	-	x	+	+	+	+	x
13	25.6076	x	x	x	x	x	x	x	x	x	-	x	x	x	x	x	-	x	-	x	+	+	+	+	x
14	26.6761	x	x	x	x	x	x	x	x	x	-	x	x	x	x	x	-	x	-	x	+	+	+	+	x
15	27.0655	x	x	x	x	x	x	x	x	x	x	-	x	x	x	x	-	x	-	x	+	+	+	+	x
16	28.6460	x	x	x	x	x	x	x	x	x	-	x	x	x	x	x	-	x	-	x	+	+	+	+	x
17	47.0649	x	x	x	x	x	x	-	x	-	-	x	x	x	x	x	x	x	x	-	+	+	+	+	x
18	48.3211	x	x	x	x	x	x	x	-	-	-	x	x	x	x	x	x	x	x	x	+	+	+	+	x
19	51.6420	x	x	x	x	x	x	x	-	-	x	-	x	x	x	x	x	x	-	x	+	+	+	+	x
20	51.8779	x	x	x	x	x	x	-	x	x	x	-	-	x	x	x	x	x	-	-	+	+	+	+	x

Table H.3: Neuron 819R10 best combinations of T3-FS4 features used for flow data linear model feature weight analysis. - denotes features not selected. x denotes features selected. + denotes features associated with hotspot locations and thus were not available for non-hotspot feature selection.

Neuron 819R32 Flow Results, Best Combinations of T3-FS4 Features Used for Linear Model Feature Weight Analysis																									
Index	TestMSE	Xconfg. hotspot contrib.	Xconfg. sin hotspot dir. * hotspot contrib.	Xconfg. cos hotspot dir. * hotspot contrib.	Oconfg. hotspot contrib.	Oconfg. sin hotspot dir. * hotspot contrib.	Oconfg. cos hotspot dir. * hotspot contrib.	sin seg. 1 dir. * seg. 1 contrib.	cos seg. 1 dir. * seg. 1 contrib.	sin seg. 2 dir. * seg. 2 contrib.	cos seg. 2 dir. * seg. 2 contrib.	sin seg. 3 dir. * seg. 3 contrib.	cos seg. 3 dir. * seg. 3 contrib.	sin seg. 4 dir. * seg. 4 contrib.	cos seg. 4 dir. * seg. 4 contrib.	sin seg. 5 dir. * seg. 5 contrib.	cos seg. 5 dir. * seg. 5 contrib.	sin seg. 6 dir. * seg. 6 contrib.	cos seg. 6 dir. * seg. 6 contrib.	sin seg. 7 dir. * seg. 7 contrib.	cos seg. 7 dir. * seg. 7 contrib.	sin seg. 8 dir. * seg. 8 contrib.	cos seg. 8 dir. * seg. 8 contrib.	sin seg. 9 dir. * seg. 9 contrib.	cos seg. 9 dir. * seg. 9 contrib.
		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	+	+	x	x	+	+	+	+	x
1	0.0093	x	x	x	x	x	x	x	x	-	x	-	x	x	x	+	+	+	+	+	+	+	+	+	+
2	0.0464	x	x	x	x	x	x	x	x	-	x	-	x	x	x	+	+	+	+	+	+	+	+	+	+
3	0.0468	x	x	x	x	x	x	x	x	-	x	-	x	x	x	+	+	+	+	+	+	+	+	+	+
4	0.0594	x	x	x	x	x	x	x	x	-	x	-	x	x	x	+	+	+	+	+	+	+	+	+	+
5	0.1165	x	x	x	x	x	x	x	x	-	x	-	x	x	x	+	+	+	+	+	+	+	+	+	+
6	0.1749	x	x	x	x	x	x	x	x	-	x	-	x	x	x	+	+	+	+	+	+	+	+	+	+
7	0.4302	x	x	x	x	x	x	x	x	-	x	-	x	x	x	+	+	+	+	+	+	+	+	+	+
8	0.4880	x	x	x	x	x	x	x	x	-	x	-	x	x	x	+	+	+	+	+	+	+	+	+	+
9	0.6333	x	x	x	x	x	x	x	x	-	x	-	x	x	x	+	+	+	+	+	+	+	+	+	+
10	0.7175	x	x	x	x	x	x	x	x	-	x	-	x	x	x	+	+	+	+	+	+	+	+	+	+
11	1.1487	x	x	x	x	x	x	x	x	-	x	-	x	x	x	+	+	+	+	+	+	+	+	+	+
12	1.1827	x	x	x	x	x	x	x	x	-	x	-	x	x	x	+	+	+	+	+	+	+	+	+	+
13	1.2272	x	x	x	x	x	x	x	x	-	x	-	x	x	x	+	+	+	+	+	+	+	+	+	+
14	1.2690	x	x	x	x	x	x	x	x	-	x	-	x	x	x	+	+	+	+	+	+	+	+	+	+
15	1.2761	x	x	x	x	x	x	x	x	-	x	-	x	x	x	+	+	+	+	+	+	+	+	+	+
16	1.4812	x	x	x	x	x	x	x	x	-	x	-	x	x	x	+	+	+	+	+	+	+	+	+	+
17	1.4817	x	x	x	x	x	x	x	x	-	x	-	x	x	x	+	+	+	+	+	+	+	+	+	+
18	1.5003	x	x	x	x	x	x	x	x	-	x	-	x	x	x	+	+	+	+	+	+	+	+	+	+
19	1.9456	x	x	x	x	x	x	x	x	-	x	-	x	x	x	+	+	+	+	+	+	+	+	+	+
20	2.0337	x	x	x	x	x	x	x	x	-	x	-	x	x	x	+	+	+	+	+	+	+	+	+	+

Table H.4: Neuron 819R32 best combinations of T3-FS4 features used for flow data linear model feature weight analysis. - denotes features not selected. x denotes features selected. + denotes features associated with hotspot locations and thus were not available for non-hotspot feature selection.

Neuron 819R64 Flow Results, Best Combinations of T3-FS4 Features Used for Linear Model Feature Weight Analysis																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									
Index	TestMSE	Xconfg. hotspot contrib.		Xconfg. sin hotspot dir. * hotspot contrib.		Xconfg. cos hotspot dir. * hotspot contrib.		Oconfg. hotspot contrib.		Oconfg. sin hotspot dir. * hotspot contrib.		Oconfg. cos hotspot dir. * hotspot contrib.		sin seg. 1 dir. * seg. 1 contrib.		cos seg. 1 dir. * seg. 1 contrib.		sin seg. 2 dir. * seg. 2 contrib.		cos seg. 2 dir. * seg. 2 contrib.		sin seg. 3 dir. * seg. 3 contrib.		cos seg. 3 dir. * seg. 3 contrib.		sin seg. 4 dir. * seg. 4 contrib.		cos seg. 4 dir. * seg. 4 contrib.		sin seg. 5 dir. * seg. 5 contrib.		cos seg. 5 dir. * seg. 5 contrib.		sin seg. 6 dir. * seg. 6 contrib.		cos seg. 6 dir. * seg. 6 contrib.		sin seg. 7 dir. * seg. 7 contrib.		cos seg. 7 dir. * seg. 7 contrib.		sin seg. 8 dir. * seg. 8 contrib.		cos seg. 8 dir. * seg. 8 contrib.		sin seg. 9 dir. * seg. 9 contrib.		cos seg. 9 dir. * seg. 9 contrib.																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									
		X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x	X	x

Table H.5: Neuron 819R64 best combinations of T3-FS4 features used for flow data linear model feature weight analysis. - denotes features not selected. x denotes features selected. + denotes features associated with hotspot locations and thus were not available for non-hotspot feature selection.



## Appendix I

### Additional Results for Linear Regression Technique Applied to Doubles Data

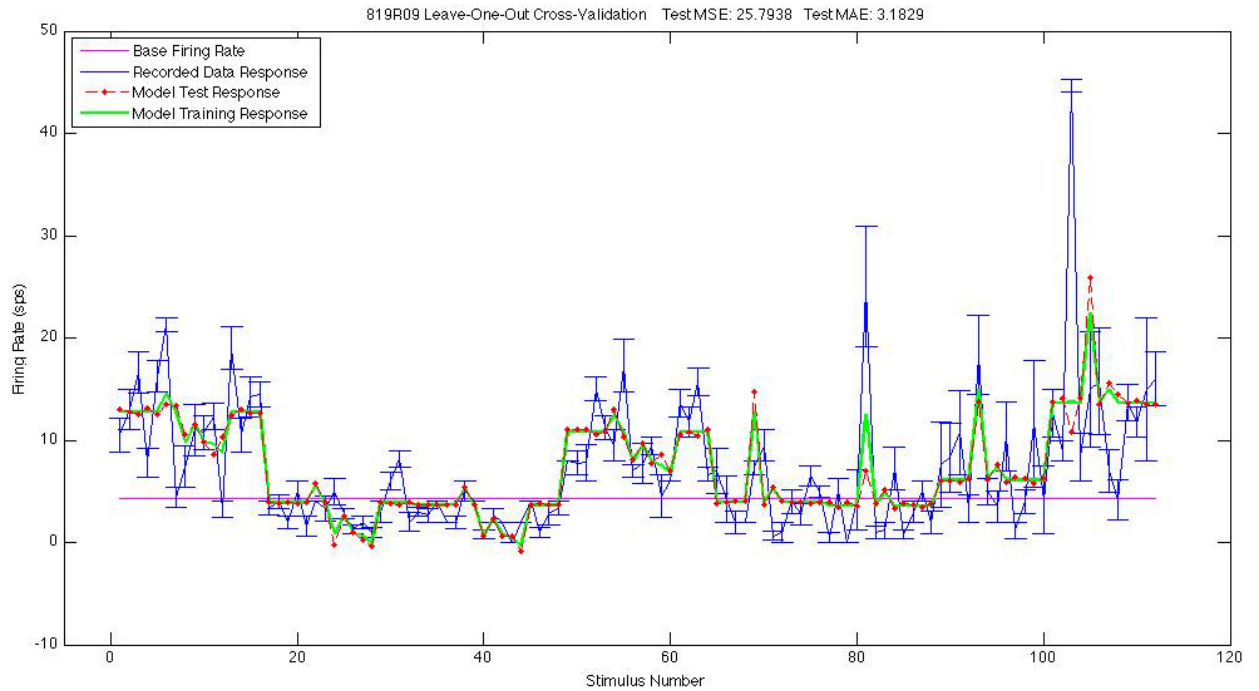


Figure I.1: Neuron 819R09 LR doubles prediction using best T3-FS4 combination.

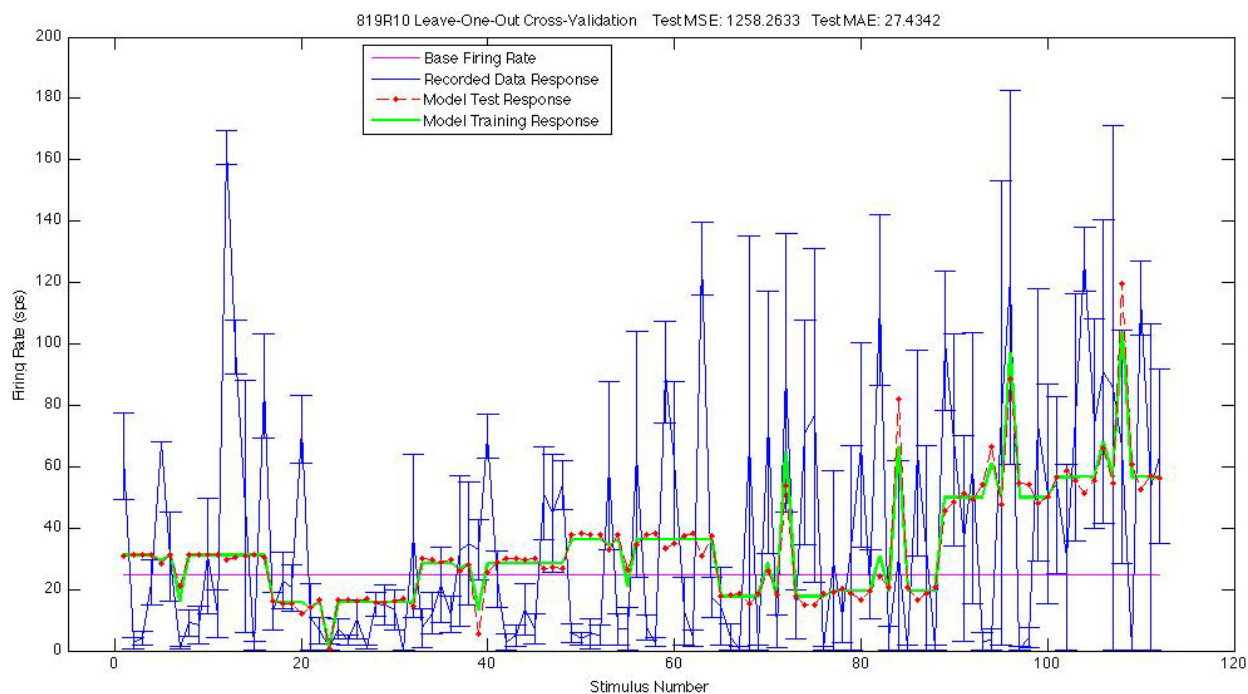


Figure I.2: Neuron 819R10 LR doubles prediction using best T3-FS4 combination.

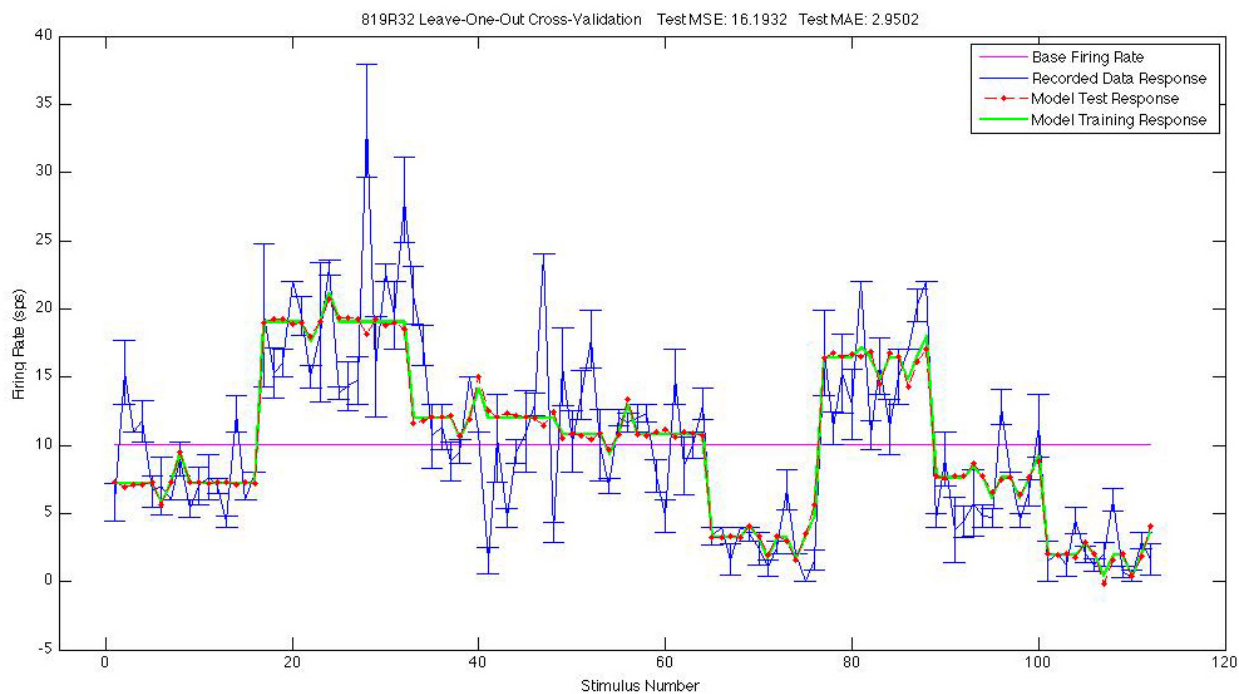


Figure I.3: Neuron 819R32 LR doubles prediction using best T3-FS4 combination.

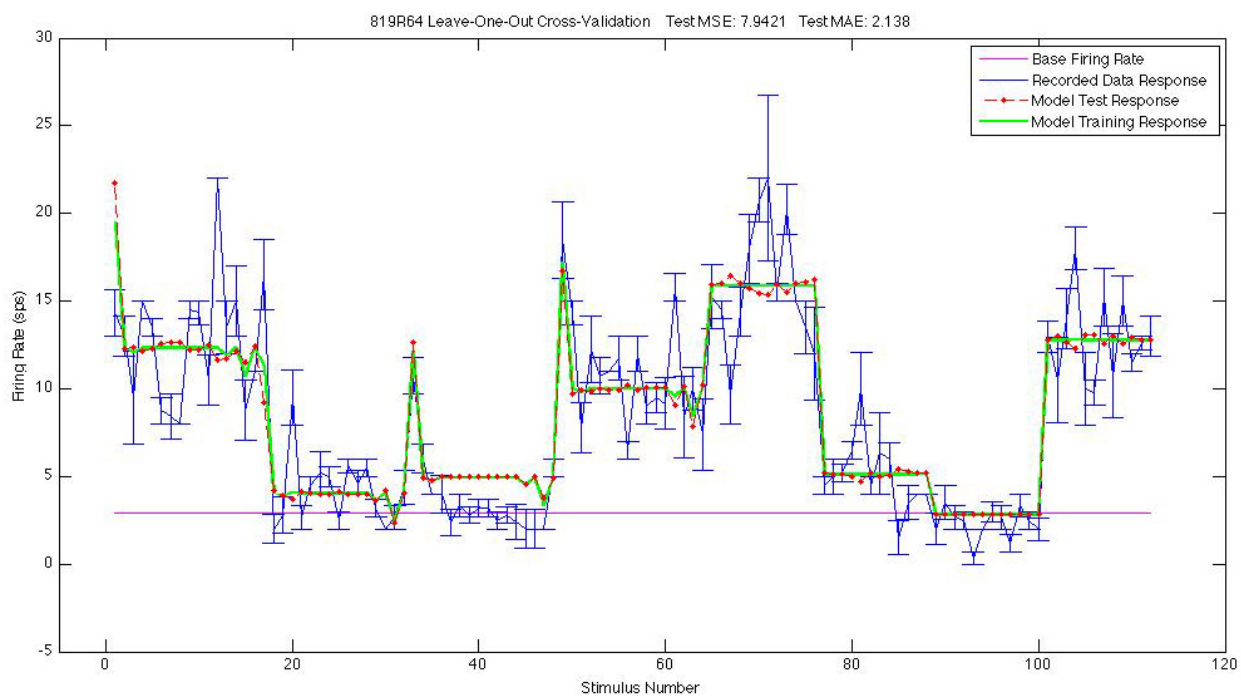


Figure I.4: Neuron 819R64 LR doubles prediction using best T3-FS4 combination.

## Appendix J

### Summary of Results by Neuron

Results Summary						
	SGMM				LR	
	Singles		Doubles		Doubles	Flow
	TrainMSE	TestMSE	TrainMSE	TestMSE	TestMSE	TestMSE
712R02	1.3060	4.9922	1.1549	0.6791	25.4990	0.0393
819R09	0.3305	5.1452	0.8867	0.4750	25.7938	0.0004
819R10	66.4277	236.2340	147.1413	16.3040	1258.2633	0.0027
819R32	0.0722	2.5213	0.5878	0.1698	16.1932	0.0093
819R64	0.1920	0.7069	0.4411	0.0642	7.9421	0.1007

Table J.1: Results summary. Singles and doubles SGMM results use the PDF as the probability function and allow for directional covariation. Doubles SGMM results are the combination of doubles subspace results. SGMM results are from random sub-sampling validation. Doubles and flow LR results shown for the best combination of T3-FS4 features. Table J.2 lists the features for each combination. LR results are from LOO-CV.

Doubles and Flow Results, Best Combinations of T3-FS4 Features																							
		Xconfg. hotspot contrib.	Xconfg. sin hotspot dir. * hotspot contrib.	Xconfg. cos hotspot dir. * hotspot contrib.	sin seg. 1 dir. * seg. 1 contrib.	cos seg. 1 dir. * seg. 1 contrib.	sin seg. 2 dir. * seg. 2 contrib.	cos seg. 2 dir. * seg. 2 contrib.	sin seg. 3 dir. * seg. 3 contrib.	cos seg. 3 dir. * seg. 3 contrib.	sin seg. 4 dir. * seg. 4 contrib.	cos seg. 4 dir. * seg. 4 contrib.	sin seg. 5 dir. * seg. 5 contrib.	cos seg. 5 dir. * seg. 5 contrib.	sin seg. 6 dir. * seg. 6 contrib.	cos seg. 6 dir. * seg. 6 contrib.	sin seg. 7 dir. * seg. 7 contrib.	cos seg. 7 dir. * seg. 7 contrib.	sin seg. 8 dir. * seg. 8 contrib.	cos seg. 8 dir. * seg. 8 contrib.	sin seg. 9 dir. * seg. 9 contrib.	cos seg. 9 dir. * seg. 9 contrib.	
Doubles	712R02	x	x	x	-	x	+	-	-	x	-	x	+	+	-	-	x	+	+	+	+	x	
	819R09	x	x	x	x	x	-	-	-	x	-	x	+	+	-	x	+	+	+	+	+	x	
	819R10	x	x	x	x	x	-	-	-	x	-	x	+	+	-	x	+	+	+	+	+	x	
	819R32	x	x	x	x	x	-	-	-	x	-	x	+	+	-	x	+	+	+	+	+	x	
	819R64	x	x	x	x	x	+	-	-	x	-	x	+	+	-	-	x	+	+	+	+	x	
Flow	712R02	x	x	x	-	x	x	x	-	x	-	x	x	-	-	x	+	+	+	+	+	x	
	819R09	x	x	x	x	x	x	x	-	x	-	x	x	-	-	x	+	+	+	+	+	x	
	819R10	x	x	x	x	x	x	x	-	x	-	x	x	-	-	x	+	+	+	+	+	x	
	819R32	x	x	x	x	x	-	-	-	x	-	x	+	+	-	x	+	+	+	+	+	x	
	819R64	x	x	x	x	x	+	-	-	x	-	x	+	+	-	-	x	+	+	+	+	x	

Table J.2: Best combinations of T3-FS4 features used for doubles and flow data prediction. - denotes features not selected. x denotes features selected. + denotes features associated with hotspot locations and thus were not available for non-hotspot feature selection. Note that the hotspots are not the same for all neurons.

## Bibliography

- [1] Genz A. An Adaptive Multidimensional Quadrature Procedure. *Communications in Computational Physics*, 4:11–15, 1972.
- [2] Paradiso M. A. Monkey Business Builds a Bridge to the Human Brain. *Nature Neuroscience*, 2:491–492, 1999.
- [3] Robinson D. A. A Method of Measuring Eye Movement Using a Scleral Search Coil in Magnetic Field. *IEEE Transactions on Biomedical Engineering*, 10:137–145, 1963.
- [4] Young R. A., Lesperance R. M., and Meyer W. W. The Gaussian Derivative Model for Spatial-Temporal Vision. *I. Cortical Model. Spatial Vision*, pages 3–4, 2001.
- [5] Chatterjee S. and Hadi A. S. Influential Observations, High Leverage Points, and Outliers in Linear Regression. *Statistical Science*, 1(3):379–416, 1986.
- [6] Dempster A. P., Laird N. M., and Rubin D. B. Discharge Patterns and Functional Organization of Mammalian Retina. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.
- [7] Draper N. and Smith H. *Applied Regression Analysis*. Wiley, 2nd edition, 1981.

- [8] Everitt B. S. and Hand D. J. *Finite Mixture Distributions*. Chapman and Hall, 1981.
- [9] Goense J. B.M., Whittingstall K., and Logothetis N. K. Functional Magnetic Resonance Imaging of Awake Behaving Macaques. *Methods*, 50(3):178–188, March 2010.
- [10] Hubel D. H. *Eye, Brain, and Vision (Scientific American Library)*. W. H. Freeman & Co (Sd), 1995.
- [11] Duffy C. J. and Wurtz R. H. Sensitivity of MST Neurons to Optic Flow Stimuli. I. A Continuum of Response Selectivity to Large-Field Stimuli. *Journal of Neurophysiology*, 65(6):1329–1345, June 1991.
- [12] Duffy C. J. and Wurtz R. H. Sensitivity of MST Neurons to Optic Flow Stimuli. II. Mechanisms of Response Selectivity Revealed by Small-Field Stimuli. *Journal of Neurophysiology*, 65(6):1346–1359, June 1991.
- [13] Logothetis N. K. The Ins and Outs of fMRI Signals. *Nature Neuroscience*, 10:1230–1232, 2007.
- [14] Stanley K. and Mäikkyläinen R. Evolving Neural Networks through Augmenting Topologies. *Evolutionary Computation*, 10, 2002.
- [15] O’Brien H. L., Tetewsky S. J., Avery L. M., Cushman L. A., Makous W., and Duffy C. J. Visual Mechanisms of Spatial Disorientation in Alzheimer’s Disease. *Cerebral Cortex*, 11(11):1083–1092, November 2001.

- [16] LeCun Y., Bottou L., Bengio Y., and Haffner P. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.
- [17] Yu C. P. Computational Model of MST Neuron Receptive Field and Interaction Effect for the Perception of Self-Motion. Master’s thesis, Rochester Institute of Technology, Rochester, New York, July 2008.
- [18] Mota S., Ros E., Díaz J., Ortigosa E. M., and Prieto A. Motion-Driven Segmentation by Competitive Neural Processing. *Neural Processing Letters*, 22(2):125–147, 2005.
- [19] Simard P. Y., Steinkraus D., and Platt J. C. Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis. In *ICDAR ’03: Proceedings of the Seventh International Conference on Document Analysis and Recognition*, page 958, Washington, DC, USA, 2003.
- [20] Steinmetz M. A., Motter B. C., Duffy C. J., and Mountcastle V. B. Functional Properties of Parietal Visual Neurons: Radial Organization of Directionalities within the Visual Field . *Journal of Neuroscience*, 7:177–191, 1987.
- [21] Kuffler S. W. Discharge Patterns and Functional Organization of Mammalian Retina. *Journal of Neurophysiology*, 16:37–68, 1955.
- [22] Yu C. P., Page W., Gaborski R., and Duffy C. Modeling the Receptive Field Organization of Optic Flow Selective MST Neurons. In *Proceedings*



*of Neuroscience 2007*, San Diego, CA, 2007.

- [23] Yu C. P., Page W., Gaborski R., and Duffy C. Modeling MST Optic Flow Responses Using Receptive Field Segmental Interactions, 2008. Poster presented as part of the proceedings of Neuroscience 2008, Washington, DC.

## Vita

Jeffrey Joseph Robble was born in Johnson City, New York, on April 22, 1985, the son of John Robble IV and Terry Robble. He received his Bachelor of Science degree in Software Engineering from the Rochester Institute of Technology, Rochester, New York in 2008. He is currently pursuing his Master of Science degree from the Rochester Institute of Technology. His research interests include Pattern Recognition, Computer Vision, Machine Learning, and Software Design Patterns. His current research interests include Neural Networks and Trend Estimation.

Permanent address: 20-1J Fairwood Drive  
Rochester, New York 14623

This thesis was typeset with L<sup>A</sup>T<sub>E</sub>X<sup>†</sup> by the author.

---

<sup>†</sup>L<sup>A</sup>T<sub>E</sub>X is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's T<sub>E</sub>X Program.